

Novell Identity Manager

3

www.novell.com

規則產生器和驅動程式自定指南

2006 年 7 月 21 日



Novell®

法律聲明

Novell, Inc. 不對本文件的內容或使用做任何陳述或保證，且特別聲明不對任何特定用途的適銷性或適用性提供任何明示或隱喻的保證。此外，Novell, Inc. 保留隨時修改本出版品及其內容的權利，且在進行此類修正或更動時，不需另行通知任何人士或公司。

此外，Novell, Inc. 不對任何軟體作任何陳述或保證，且特別聲明不對任何特定用途的適銷性或適用性提供任何明示或隱喻的保證。此外，Novell, Inc. 保留隨時修改任何或全部 Novell 軟體的權利，且在進行此類更動時，不需通知任何人士或公司。

這份授權書中所提及的任何產品或技術資訊皆受到美國出口管制法 (U.S. Export Control) 及其他國家的交易法約束。您同意遵守所有出口管制法規，並取得出口、再出口或進口交付物品所需之任何必要的授權或類別。您同意不出口或再出口至目前美國出口排除清單上所列公司，或者至美國出口法所指定之禁運或恐怖份子的國家。您同意不將交付產品用在禁止的核子武器、飛彈或化學生物武器等用途上。如需更詳細的 Novell 軟體出口資訊，請參閱 www.novell.com/info/exports/。Novell 無須承擔您無法取得任何必要的出口核准之責任。

版權 © 2005 Novell, Inc. 版權所有。未經出版者的書面同意，本出版品的任何部份皆不可複製、影印、傳送，或是儲存在可擷取系統上。

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.
www.novell.com

線上文件：若要存取本產品及其他 Novell 產品的線上文件，或取得更新，請參閱 www.novell.com/documentation。

Novell 商標

DirXML 是 Novell, Inc. 在美國與其他國家的註冊商標。

eDirectory 是 Novell, Inc. 的商標。

Novell 是 Novell, Inc. 在美國與其他國家的註冊商標。

Nsure 是 Novell, Inc. 的商標。

協力廠商資料

所有的協力廠商商標均為其個別擁有廠商的財產。

目錄

關於本指南	7
1 規則和過濾器	9
1.1 什麼是規則和過濾器？	9
1.1.1 舊版術語的變更	10
1.1.2 DirXML 程序檔	11
1.2 規則介紹	11
1.2.1 規則	11
1.2.2 定義規則	31
1.3 過濾器	32
2 搭配 Designer 使用規則產生器定義規則	33
2.1 規則	33
2.2 Designer 中的規則產生器任務	34
2.2.1 開啓規則產生器	34
2.2.2 建立規則	38
2.2.3 建立規則	46
2.2.4 建立引數	55
2.2.5 編輯規則	65
2.2.6 使用預先定義的規則	68
2.2.7 使用「規則模擬器」測試規則	97
2.2.8 編輯 DirXML 程序檔	106
2.3 一般運算式	112
2.4 XPath 1.0 運算式	113
2.5 條件	114
2.5.1 If 關聯	114
2.5.2 If Attribute	115
2.5.3 If Class Name	116
2.5.4 If Destination Attribute	117
2.5.5 If 目的 DN	118
2.5.6 If Entitlement	119
2.5.7 If Global Configuration Value	120
2.5.8 If Local Variable	121
2.5.9 If Named Password	123
2.5.10 If Operation	123
2.5.11 If Operation Attribute	125
2.5.12 If Operation Property	126
2.5.13 If Password	127
2.5.14 If Source Attribute	127
2.5.15 If Source DN	128
2.5.16 If XPath Expression	129
2.6 動作	130
2.6.1 新增關聯	131
2.6.2 新增目的屬性值	132
2.6.3 新增目的物件	133
2.6.4 新增來源屬性值	135
2.6.5 新增來源物件	136
2.6.6 附加 XML 元素	136
2.6.7 附加 XML 文字	137

2.6.8	中斷	138
2.6.9	清除目的屬性值	138
2.6.10	清除操作內容	139
2.6.11	清除來源屬性值	139
2.6.12	清除單次登入 (SSO) 身分證明	140
2.6.13	由 XPATH 運算式複製	140
2.6.14	複製操作屬性	141
2.6.15	刪除目的物件	142
2.6.16	刪除來源物件	143
2.6.17	尋找相符物件	143
2.6.18	針對每個	144
2.6.19	產生事件	145
2.6.20	實作授權	147
2.6.21	移動目的物件	148
2.6.22	移動來源物件	149
2.6.23	重新格式化操作屬性	150
2.6.24	移除關聯	151
2.6.25	移除目的屬性值	152
2.6.26	移除來源屬性值	152
2.6.27	重新命名目的物件	153
2.6.28	重新命名操作屬性	154
2.6.29	重新命名來源物件	154
2.6.30	傳送電子郵件	155
2.6.31	從範本傳送電子郵件	156
2.6.32	設定預設的屬性值	157
2.6.33	設定目的屬性值	158
2.6.34	設定目的密碼	159
2.6.35	設定本地變數	160
2.6.36	設定操作關聯	161
2.6.37	設定操作類別名稱	162
2.6.38	設定操作目的 DN	162
2.6.39	設定操作內容	163
2.6.40	設定操作來源 DN	164
2.6.41	設定操作範本 DN	164
2.6.42	設定來源屬性值	165
2.6.43	設定來源密碼	166
2.6.44	設定 SSO 身分證明	166
2.6.45	設定 SSO 通關密語	167
2.6.46	設定 XML 屬性	168
2.6.47	狀態	168
2.6.48	條紋化操作屬性	169
2.6.49	條紋化 XPath	170
2.6.50	追蹤訊息	170
2.6.51	否決	171
2.6.52	若無法使用操作屬性則否決	172
2.7	名詞記號	173
2.7.1	新增的授權	173
2.7.2	關聯	174
2.7.3	屬性	174
2.7.4	類別名稱	175
2.7.5	目的屬性	175
2.7.6	目的 DN	176
2.7.7	目的名稱	177
2.7.8	授權	177
2.7.9	全域組態值	178
2.7.10	本地變數	178
2.7.11	具名密碼	179
2.7.12	操作	179

2.7.13	操作屬性	180
2.7.14	操作內容	180
2.7.15	密碼	181
2.7.16	移除的屬性	181
2.7.17	移除的授權	181
2.7.18	來源屬性	181
2.7.19	來源 DN	182
2.7.20	來源名稱	182
2.7.21	文字	183
2.7.22	唯一名稱	184
2.7.23	Unmatched Source DN	185
2.7.24	XPath	186
2.8	動詞記號	186
2.8.1	逸出目的 DN	187
2.8.2	逸出來源 DN	187
2.8.3	小寫	187
2.8.4	剖析 DN	188
2.8.5	取代全部	190
2.8.6	取代第一個	191
2.8.7	子字串	192
2.8.8	大寫	193
2.9	值	194
2.9.1	比較模式	194
3	使用 iManager 內的規則產生器定義規則	197
3.1	規則	197
3.2	iManager 中的規則產生器任務	198
3.2.1	開啓「規則產生器」	198
3.2.2	建立規則	198
3.2.3	在規則 (Policy) 內定義各別的規則 (Rule)	199
3.2.4	在規則 (Ruley) 內定義各別的引數	200
3.2.5	修改規則	207
3.2.6	移除規則	207
3.2.7	重新命名規則	208
3.2.8	刪除規則	208
3.2.9	從 XML 檔案輸入規則	208
3.2.10	將規則輸入至 XML 檔案	208
3.2.11	建立規則參考	209
3.2.12	使用預先定義的規則	209
3.3	一般運算式	229
3.4	XPath 1.0 運算式	229
3.5	條件	230
3.5.1	If 關聯	230
3.5.2	If Attribute	231
3.5.3	If 類別名稱	232
3.5.4	If Destination Attribute	233
3.5.5	If 目的 DN	234
3.5.6	If Entitlement	235
3.5.7	If Global Configuration Value	236
3.5.8	If Local Variable	237
3.5.9	If Named Password	239
3.5.10	If 操作	239
3.5.11	If Operation Attribute	241
3.5.12	If 操作內容	242
3.5.13	If Password	243
3.5.14	If Source Attribute	243

	3.5.15	If Source DN	245
	3.5.16	If XPath Expression	246
3.6		動作	247
	3.6.1	新增關聯	248
	3.6.2	新增目的屬性值	249
	3.6.3	新增目的物件	250
	3.6.4	新增來源屬性值	251
	3.6.5	新增來源物件	252
	3.6.6	附加 XML 元素	253
	3.6.7	附加 XML 文字	253
	3.6.8	中斷	254
	3.6.9	清除目的屬性值	254
	3.6.10	清除操作內容	255
	3.6.11	清除單次登入 (SSO) 身分證明	255
	3.6.12	清除來源屬性值	256
	3.6.13	由 XPath 運算式複製	256
	3.6.14	複製操作屬性	257
	3.6.15	刪除目的物件	257
	3.6.16	刪除來源物件	258
	3.6.17	尋找相符物件	258
	3.6.18	針對每個	259
	3.6.19	產生事件	260
	3.6.20	實作授權	262
	3.6.21	移動目的物件	263
	3.6.22	移動來源物件	264
	3.6.23	重新格式化操作屬性	265
	3.6.24	移除關聯	265
	3.6.25	移除目的屬性值	266
	3.6.26	移除來源屬性值	267
	3.6.27	重新命名目的物件	267
	3.6.28	重新命名操作屬性	268
	3.6.29	重新命名來源物件	268
	3.6.30	傳送電子郵件	269
	3.6.31	從範本傳送電子郵件	270
	3.6.32	設定預設的屬性值	271
	3.6.33	設定目的屬性值	272
	3.6.34	設定目的密碼	273
	3.6.35	設定本地變數	274
	3.6.36	設定操作關聯	275
	3.6.37	設定操作類別名稱	275
	3.6.38	設定操作目的 DN	275
	3.6.39	設定操作內容	276
	3.6.40	設定操作來源 DN	276
	3.6.41	設定操作範本 DN	277
	3.6.42	設定來源屬性值	277
	3.6.43	設定來源密碼	278
	3.6.44	設定 SSO 身分證明	279
	3.6.45	設定 SSO 通關密語	279
	3.6.46	設定 XML 屬性	280
	3.6.47	設定 SSO 身分證明	281
	3.6.48	狀態	282
	3.6.49	去除操作屬性	282
	3.6.50	去除 XPath	283
	3.6.51	追蹤訊息	283
	3.6.52	否決	284
	3.6.53	若無法使用操作屬性則否決	284
3.7		名詞記號	285
	3.7.1	新增的授權	286

3.7.2	關聯	286
3.7.3	屬性	287
3.7.4	類別名稱	287
3.7.5	目的屬性	287
3.7.6	目的 DN	288
3.7.7	目的名稱	289
3.7.8	授權	290
3.7.9	全域組態值	290
3.7.10	本地變數	290
3.7.11	具名密碼	291
3.7.12	操作	292
3.7.13	操作屬性	292
3.7.14	操作內容	293
3.7.15	密碼	293
3.7.16	移除的屬性	293
3.7.17	Removed Entitlements	293
3.7.18	Source Attribute	294
3.7.19	Source DN	294
3.7.20	Source Name	295
3.7.21	Text	295
3.7.22	Unique Name	296
3.7.23	Unmatched Source DN	297
3.7.24	XPath	298
3.8	動詞記號	298
3.8.1	Escape Destination DN	299
3.8.2	Escape Source DN	299
3.8.3	Lower Case	299
3.8.4	Parse DN	300
3.8.5	Replace All	302
3.8.6	Replace First	302
3.8.7	Substring	304
3.8.8	Upper Case	305
3.9	值	305
3.9.1	比較模式	305
4	Novell 身分證明提供規則	307
4.1	具有 Novell SecureLogin 的身分證明提供規則	307
4.2	實作具有 SecureLogin 的身分證明提供規則	309
4.2.1	符合具有 Novell SecureLogin 之身分證明提供規則的要求	309
4.2.2	延伸 Novell SecureLogin 的 LDAP 綱要	309
4.2.3	決定 Novell SecureLogin 的部署組態參數	310
4.2.4	建立 Novell SecureLogin 的儲存機制物件	312
4.2.5	建立 Novell SecureLogin 的應用程式物件	319
4.2.6	設定 Novell SecureLogin 的身分證明提供規則組態	325
4.3	具有 Novell SecretStore 的身分證明提供規則	328
4.4	實作具有 SecretStore 的身分證明提供規則	330
4.4.1	符合具有 Novell SecretStore 之身分證明提供規則的要求	330
4.4.2	決定 Novell SecretStore 的部署組態參數	331
4.4.3	建立 Novell SecretStore 的儲存機制物件	333
4.4.4	建立 Novell SecretStore 的應用程式物件	340
4.4.5	設定 Novell SecretStore 的身分證明提供規則組態	346
5	使用 XSLT 樣式表定義規則	351
5.1	在 Designer 中管理 XSLT 樣式表	351
5.1.1	在 Designer 中新增 XSLT 規則	351

5.2	在 iManager 中管理 XSLT 樣式表	353
5.2.1	在 iManager 中新增 XSLT 規則	353
5.3	開始使用身分轉換	354
5.4	使用 Identity Manager 傳遞的參數	354
5.5	使用延伸功能	356
5.6	建立密碼範例：建立規則	357
5.7	建立 eDirectory 使用者範例：建立規則	358
6	管理過濾器	365
6.1	Designer 中的過濾器任務	365
6.1.1	存取過濾器編輯器	365
6.1.2	編輯過濾器	368
6.1.3	測試過濾器	372
6.1.4	檢視過濾器的 XML 來源	378
6.1.5	其他過濾器選項	383
6.2	iManager 中的過濾器任務	385
6.2.1	存取過濾器	385
6.2.2	編輯過濾器	385
7	管理綱要映射規則	389
7.1	Designer 中的綱要映射規則任務	389
7.1.1	存取綱要映射編輯器	389
7.1.2	編輯綱要映射規則	393
7.1.3	測試綱要映射規則	396
7.1.4	存取綱要映射規則 XML	402
7.1.5	綱要映射規則的其他選項	408
7.2	iManager 中的綱要映射規則任務	411
7.2.1	存取綱要映射規則	412
7.2.2	編輯綱要映射規則	412

關於本指南

Novell® Identity Manager 3.0® 是一種資料共享和同步服務，可讓應用程式、目錄和資料庫共享資訊。它會將散佈的資訊連結在一起，並讓您建立規則，用於在身分發生變更時管理對指定系統的自動更新。

Identity Manager 提供了下列項目的基礎：帳戶提供、安全性、單次登入、使用者自助服務、驗證、授權、自動工作流程和 Web 服務。它可讓您整合、管理和控制您散佈的身分資訊，以便安全地將正確的資源傳送給正確的人員。

本指南提供 Identity Manager 3.0 中「規則產生器」和「驅動程式組態」的詳細參考資訊。

- ◆ 第 1 章 「規則和過濾器」, 第 9 頁
- ◆ 第 2 章 「搭配 Designer 使用規則產生器定義規則」, 第 33 頁
- ◆ 第 3 章 「使用 iManager 內的規則產生器定義規則」, 第 197 頁
- ◆ 第 5 章 「使用 XSLT 樣式表定義規則」, 第 351 頁
- ◆ 第 6 章 「管理過濾器」, 第 365 頁
- ◆ 第 7 章 「管理綱要映射規則」, 第 389 頁

使用對象

本指南適用於 Identity Manager 3.0 管理員。

意見反應

我們想知道您對於本手冊與其他本產品隨附之文件的意見與建議。請使用線上文件中每頁底下的「使用者意見」功能，或造訪 www.novell.com/documentation/feedback.html，然後寫下您的意見。

文件更新

如需本文件的最新版本，請參閱 [Identity Manager 文件網站 \(http://www.novell.com/documentation/idm\)](http://www.novell.com/documentation/idm)

如需 Identity Manager 2.0 的相關文件，請參閱 [Identity Manager 文件網站 \(http://www.novell.com/documentation/idm\)](http://www.novell.com/documentation/idm)

其他文件

如需使用 Identity Manager 驅動程式的相關文件，請參閱 [Identity Manager 文件網站 \(http://www.novell.com/documentation/idmdrivers/index.html\)](http://www.novell.com/documentation/idmdrivers/index.html)

文件慣例

本文件中使用大於符號 (>) 分隔步驟中的各個動作，以及前後參照路徑中的數個項目。

商標符號 (®、™ 等) 代表 Novell 的商標。星號 (*) 代表協力廠商的商標。

規則和過濾器

本節包含規則和過濾器的綜覽，以及它們在 Identity Manager 環境的功能。涵蓋的主題如下：

- ◆ 「什麼是規則和過濾器？」，第 9 頁
- ◆ 「規則介紹」，第 11 頁

1.1 什麼是規則和過濾器？

在高層級中，規則可讓您自定 Identity Manager 傳送和接收更新的方式。

對驅動程式 Shim 撰寫目的之細節若有某種程度的瞭解，將有助於瞭解規則。

當撰寫驅動程式 Shim 時，會嘗試將公司部署驅動程式時，可能會使用的所有項目同步化的能力包含在內。開發人員會撰寫驅動程式 Shim 來偵測已連接系統中的任何相關變更，然後將此變更傳送至 Identity Vault。

此變更包含於 XML 文件中，並根據 Identity Manager 的規格格式化。以下片段包含一份此類的 XML 文件：

```
<nds dtdversion="2.0" ndsversion="8.7.3">
<source>
  <product version="2.0">DirXML</product>
  <contact>Novell, Inc.</contact>
</source>

<input>
  <add class-name="User" event-id="0" src-dn="\ACME\Sales\Smith"
  src-entry-id="33071">
    <add-attr attr-name="Surname">
      <value timestamp="1040071990#3" type="string">Smith</value>
    </add-attr>
    <add-attr attr-name="Telephone Number">
      <value timestamp="1040072034#1" type="teleNumber">111-1111</
value>
    </add-attr>
  </add>
</input>
</nds>
```

設計驅動程式是爲了報告所有的相關變更，讓您能夠過濾資訊。設計過濾器是爲了封鎖資訊。您可以修改過濾器，只讓所需的資訊進入您的環境。判斷什麼變更是重要的，以及如何處理這些變更是在引擎中進行的，而不是在驅動程式 Shim 中處理的。

如果公司不是很在意群組，則他們可以實作過濾器來封鎖 Identity Vault 或已連接系統中所有與群組相關的操作。如果公司在意使用者和群組，則他們可以實作過濾器，讓兩種物件類型在 Identity Vault 與已連接系統之間同步化。

驅動程式自定的第一個步驟，是將過濾器定義爲只允許同步化您感興趣的物件。

下一個步驟則定義 **Identity Manager** 要對您的過濾器允許的物件進行什麼動作。請參考上述 XML 文件中的新增操作做為範例。有一個使用者名為 **Smith**，其電話號碼為 111-1111，被加入了您的已連接系統。假設您允許此操作，**Identity Manager** 則需決定要對這個使用者做什麼。

為了做出此決定，**Identity Manager** 會依特定的順序套用規則集。

首先，「相符」規則會回答「此物件是否已在資料儲存中？」這個問題。若要回答這個問題，您需要定義物件的獨特特性。一般會檢查的屬性可能是電子郵件地址，因為這些資訊通常是獨一無二的。您可以定義這樣的規則：「如果兩個物件有相同的電子郵件地址，則它們是相同的物件」。

如果找到相符的項目，**Identity Manager** 會將它記錄在一個名為關聯的屬性中。關聯是指讓 **Identity Manager** 將已連接系統中的物件互相關聯的獨特值。

在找不到相符項的情況下，則會呼叫「建立」規則。「建立」規則會告訴 **Identity Manager** 您想要在什麼情況下建立物件。您可以指定在建立規則 (**Rule**) 中必須出現的特定屬性。如果沒有這些屬性，**Identity Manager** 會在必要資訊提供之前，防止物件的建立。

在物件建立後，「佈置」規則會告訴 **Identity Manager** 要將物件放在哪裡。您可以指定物件應該建立在與它們來源系統相同的階層式結構中，也可以根據屬性值，將它們放置在完全不同的位置。

如果您要根據物件的位置屬性，將使用者放置在階層中，並根據 **Full Name** 為它們命名，可以在建立規則中將這些屬性設定為必要。這可以確保屬性會存在，讓您的佈置策略可以正確運作。

您還可以利用規則來執行許多其他的事項。使用「規則產生器」，您可以輕易產生唯一值、新增和移除屬性、產生事件與指令、傳送電子郵件等等。使用 **XSLT** 直接轉換 XML 文件可進行數種更為進階的轉換作業（請記住，送出和送入 **Identity Vault** 的變更都是以 XML 文件進行的）。

您要記住的基本事項是：規則可讓您控制 **Identity Manager** 處理更新的方式。

繼續瀏覽至「規則介紹」，第 11 頁以瞭解其他規則類型的相關資訊，然後移至第 2 章「搭配 **Designer** 使用規則產生器定義規則」，第 33 頁或第 3 章「使用 **iManager** 內的規則產生器定義規則」，第 197 頁以瞭解如何使用「規則產生器」。

1.1.1 舊版術語的變更

在 **DirXML® 1.1a** 中，會使用詞彙「規則 (**Rule**)」來描述規則集、此規則集中的個別規則，以及個別規則中的條件和動作，視上下文而定。在上下文不明確的情況下，此重疊會造成混淆。

目前在 **Identity Manager 2** 中，則使用詞彙「規則 (**Policy**)」取代先前使用的詞彙「規則 (**Rule**)」，來描述所發生的高層級轉換。您現在可以定義規則 (**Policy**) 集，其中每個規則 (**Policy**) 都包含一或多個規則 (**Rule**)。而詞彙「規則 (**Rule**)」現在只用來描述個別的條件和動作集。

下表顯示從 **DirXML 1.1a** 到 **Identity Manager 2.x** 的術語變更。

表格 1-1 從 DirXML 1.1a 到 Identity Manager 2.x 的術語變更

概念	DirXML 1.1a 術語	Identity Manager 2.x 術語
轉換集	規則 (Rule)	規則 (Policy) 集
轉換集中的個別轉換	規則 (Rule)	規則 (Policy)
個別轉換中的條件和動作	規則 (Rule)	規則 (Rule)

下表顯示從 Identity Manager 2.x 到 Identity Manager 3.0 的術語變更。

表格 1-2 從 Identity Manager 2.x 到 Identity Manager 3.0 的術語變更

概念	Identity Manager 2.x 術語	Identity Manager 3 術語
產品	DirXML	Identity Manager
已安裝產品的伺服器	DirXML 伺服器	Metadirectory 伺服器
應用程式或資料庫中進行資料同步化的伺服器	DirXML 已連接系統伺服器	已連接系統伺服器
儲存物件的位置	eDirectory™	Identity Vault
處理用元件	DirXML 引擎	Metadirectory 引擎

1.1.2 DirXML 程序檔

「DirXML 程序檔」是實作 Identity Manager 規則的主要方法。它描述透過有序規則 (Rule) 集實作的規則 (Policy)。規則 (Rule) 是由要測試的一組條件，以及當條件符合時要執行的一組有序動作組合而成。

「DirXML 程序檔」是使用「規則產生器」建立的。「規則產生器」提供了容易使用的 GUI 介面。

1.2 規則介紹

本節介紹可用的規則類型、它們在 Identity Manager 中的角色，以及如何定義您自己的規則。涵蓋的主題如下：

- ◆ 「規則」，第 11 頁
- ◆ 「定義規則」，第 31 頁

1.2.1 規則

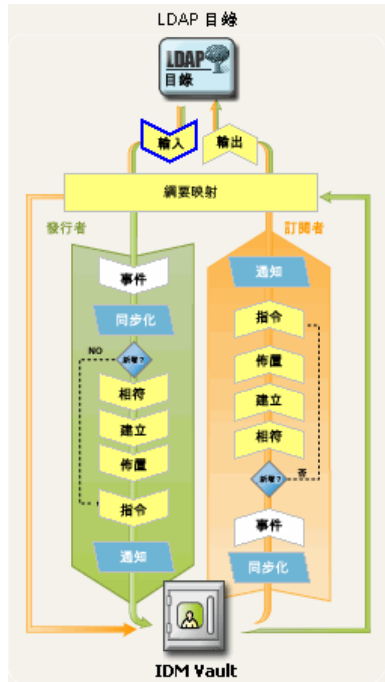
您可以在「訂閱者」和「發行者」通道上定義數種不同的規則類型。每個規則會在不同的資料轉換步驟中套用，而有些規則只有在特定動作出現時才會套用。例如，「建立」規則只有在建立新物件時才會套用。

規則在通道上的執行順序是：

- ◆ 「事件轉換規則」，第 12 頁

- ◆ 「相符規則」，第 15 頁
- ◆ 「建立規則」，第 16 頁
- ◆ 「佈置規則」，第 19 頁
- ◆ 「指令轉換規則」，第 22 頁
- ◆ 「綱要映射規則」，第 25 頁
- ◆ 「輸出轉換規則」，第 27 頁
- ◆ 「輸入轉換規則」，第 29 頁

特性 1-1 規則的執行順序



事件轉換規則

「事件轉換」規則會改變 Metadirectory 引擎對於 Identity Vault 或已連接應用程式中發生的事件之檢視方式。「事件轉換」規則中最常執行的任務是自定過濾，例如範圍過濾和事件類型過濾。

範圍過濾會根據事件位置或屬性值，移除不想要的事件。比方說，如果部門屬性不等於特定的值或不是特定群組的成員，就將事件移除。

事件類型過濾會根據事件類型，移除不想要的事件。例如，移除所有刪除事件。

範例：

- ◆ 範圍過濾
- ◆ 類型過濾

範圍過濾：此範例「DirXML 程序檔」規則只讓下列使用者的事件通過：包含在「使用者」子網路樹中、未被停用，且在 Title 屬性中不包含 Consultant 或 Manager 一字。它也會產生一個狀態文件，指出操作被封鎖的時間。


```

<policy>
  <rule>
    <description>Scope Filtering</description>
    <conditions>
      <or>
        <if-class-name op="equal">User</if-class-name>
      </or>
      <or>
        <if-src-dn op="not-in-subtree">Users</if-
src-dn>
        <if-attr name="Login Disabled"
op="equal">True</if-attr>
        <if-attr mode="regex" name="Title"
op="equal">.*Consultant.*</if-attr>
        <if-attr mode="regex" name="Title"
op="equal">.*Manager.*</if-attr>
      </or>
    </conditions>
    <actions>
      <do-status level="error">
        <arg-string>
          <token-text>User doesn't meet required
conditions</token-text>
          </arg-string>
        </do-status>
      <do-veto/>
    </actions>
  </rule>
</policy>

```

此「DirXML 程序檔」規則允許在「使用者」物件上進行修改操作，但是不能修改已相關聯的物件。

```

<policy>
  <rule>
    <description>Veto all operation on User except modifies
of already associated objects</description>
    <conditions>
      <or>
        <if-class-name op="equal">User</if-class-name>
      </or>
      <or>
        <if-operation op="not-equal">modify</if-
operation>
        <if-association op="not-associated"/>
      </or>
    </conditions>
    <actions>
      <do-veto/>
    </actions>
  </rule>

```

```
</policy>
```

類型過濾 - 此「DirXML 程序檔」規則 (Policy) 的第一個規則 (Rule) 只允許同步化 Employee 和 Contractor 容器中的物件。第二個規則 (Rule) 阻擋了所有「重新命名」和「移動」操作。

```
<policy>
  <rule>
    <description>Only synchronize the Employee and Contractor
subtrees</description>
    <conditions>
      <and>
        <if-src-dn op="not-in-
container">Employees</if-src-dn>
        <if-src-dn op="not-in-
container">Contractors</if-src-dn>
      </and>
    </conditions>
    <actions>
      <do-status level="warning">
        <arg-string>
          <token-text>Change ignored: Out of
scope.</token-text>
        </arg-string>
      </do-status>
    </do-veto/>
    </actions>
  </rule>
  <rule>
    <description>Don't synchronize moves or renames</
description>
    <conditions>
      <or>
        <if-operation op="equal">move</if-
operation>
        <if-operation op="equal">rename</if-
operation>
      </or>
    </conditions>
    <actions>
      <do-status level="warning">
        <arg-string>
          <token-text>Change ignored:
We don't like you to do that.</token-text>
        </arg-string>
      </do-status>
    </do-veto/>
    </actions>
  </rule>
</policy>
```

此「DirXML 程序檔」規則會封鎖所有「新增」事件。

```

<policy>
  <rule>
    <description>Type Filtering</description>
    <conditions>
      <and>
        <if-operation op="equal">add</if-
operation>
      </and>
    </conditions>
    <actions>
      <do-status level="warning">
        <arg-string>
          <token-text>Change ignored:
Adds are not allowed.</token-text>
        </arg-string>
      </do-status>
      <do-veto/>
    </actions>
  </rule>
</policy>

```

相符規則

「相符」規則 (例如「訂閱者相符」和「發行者相符」) 會在目的資料儲存內，尋找與來源資料儲存中未關聯物件對應的物件。請務必注意，「相符」規則並非一定需要。

例如，以下情況可能就不需要「相符」規則：

- ◆ 在沒有已存在或對應的物件時執行啓始移轉。

「相符」規則必須小心制訂，確保「相符」規則不會找到不正確的相符項目。

範例：

- ◆ 依據 Internet Email Address 比對
- ◆ 依據 Common Name 比對

依據 ID 比對：此範例「DirXML 程序檔」規則會依據 Internet Email Address 比對使用者。

```

<policy>
  <rule>
    <description>Match Users based on email address</
description>
    <conditions>
      <and>
        <if-class-name op="equal">User</if-class-name>
      </and>
    </conditions>
    <actions>
      <do-find-matching-object>
        <arg-dn>
          <token-text>ou=people,o=novell</token-text>
        </arg-dn>
      </do-find-matching-object>
    </actions>
  </rule>
</policy>

```

```

        <arg-match-attr name="Internet EMail Address"/>
        </do-find-matching-object>
    </actions>
</rule>
</policy>

```

依據名稱比對：此範例「DirXML 程序檔」規則會依據其 Common Name 屬性比對「群組」物件。

```

<?xml version="1.0" encoding="UTF-8"?>
<policy>
    <rule>
        <description>Match Group by Common Name</description>
        <conditions>
            <or>
                <if-class-name op="equal">Group</
if-class-name>
            </or>
        </conditions>
        <actions>
            <do-find-matching-object scope="subtree">
                <arg-match-attr name="CN"/>
            </do-find-matching-object>
        </actions>
    </rule>
</policy>

```

建立規則

「建立」規則（例如「訂閱者建立」規則和「發行者建立」規則）定義了建立新物件必須符合的條件。沒有「建立」規則表示可以建立物件。

例如，您在 Identity Vault 中建立新使用者，但是您只提供名稱和 ID 給新的「使用者」物件。此建立動作會在 eDirectory 網路樹中鏡像複製，但是由於您有一個「建立」規則，指定只允許有更完整定義的「使用者」物件，所以此新增不會立即反應在已連接至 Identity Vault 的應用程式。

「發行者」通道和「訂閱者」通道的「建立」規則可以相同，也可以不同。

要在 eDirectory 中建立物件時，可以指定要在建立程序中使用「範本」物件。

「建立」規則常用來進行以下動作：

- ◆ 否決建立條件不符的物件（可能是因為沒有屬性）。
- ◆ 提供預設的屬性值。
- ◆ 提供預設的密碼。

範例：

- ◆ 必要屬性
- ◆ 預設的屬性值
- ◆ 預設的密碼

◆ 指定範本

必要屬性：此範例「DirXML 程序檔」規則 (Policy) 的第一個規則 (Rule) 要求「使用者」物件需包含 CN、Given Name、Surname 和 Internet EMail Address 屬性，才能建立使用者。第二個規則 (Rule) 要求所有「組織單位」物件都要有 OU 屬性。最後一個規則 (Rule) 否決了所有名稱爲 Fred 的「使用者」物件。

```
<policy>
  <rule>
    <description>Veto if required attributes CN, Given Name,
Surname and Internet EMail Address not available</description>
    <conditions>
      <or>
        <if-class-name op="equal">User</if-class-
name>
          </or>
        </or>
      </conditions>
    <actions>
      <do-veto-if-op-attr-not-available name="CN"/>
      <do-veto-if-op-attr-not-available name="Given Name"/>
      <do-veto-if-op-attr-not-available name="Surname"/>
      <do-veto-if-op-attr-not-available name="Internet
EMail Address"/>
    </actions>
  </rule>
  <rule>
    <description>Organizational Unit Required Attributes</
description>
    <conditions>
      <or>
        <if-class-name op="equal">Organizational
Unit</if-class-name>
          </or>
        </or>
      </conditions>
    <actions>
      <do-veto-if-op-attr-not-available name="OU"/>
    </actions>
  </rule>
</policy>
```

預設的屬性值：此範例「DirXML 程序檔」規則會爲使用者的 Description 屬性加入預設值。

```
<policy>
  <rule>
    <description>Default Description of New Employee</
description>
    <conditions>
      <or>
        <if-class-name op="equal">User</if-class-name>
          </or>
        </or>
      </conditions>
    <actions>
```

```

        <do-set-default-attr-value name="Description">
            <arg-value type="string">
                <token-text>New Employee</token-text>
            </arg-value>
        </do-set-default-attr-value>
    </actions>
</rule>
</policy>

```

預設的密碼：此範例「DirXML 程序檔」規則提供密碼值的建立，密碼值是以名字的前兩個字元和姓氏的前六個字元組成，均為小寫。

```

<policy>
    <rule>
        <description>Default Password of [2]FN+[6]LN</
description>
        <conditions>
            <and>
                <if-class-name op="equal">User</if-class-name>
                <if-password op="not-available"/>
            </and>
        </conditions>
        <actions>
            <do-set-dest-password>
                <arg-string>
                    <token-lower-case>
                        <token-substring length="2">
                            <token-op-attr name="Given
Name"/>
                                </token-substring>
                            <token-substring length="6">
                                <token-op-attr
name="Surname"/>
                                    </token-substring>
                                </token-lower-case>
                            </arg-string>
                        </do-set-dest-password>
                    </actions>
                </rule>
            </policy>

```

指定範本：如果使用者的 Title 屬性指出使用者是 Manager (包含 "Manager")，此範例「DirXML 程序檔」規則會指定一個範本物件。

```

<policy>
    <rule>
        <description>Assign Manager Template if Title
contains Manager</description>
        <conditions>
            <and>
                <if-class-name op="equal">User</if-class-
name>

```

```

        <if-op-attr name="Title" op="available"/
    >
        <if-op-attr mode="regex" name="Title"
op="equal">.*Manager.*</if-op-attr>
        </and>
    </conditions>
    <actions>
        <do-set-op-template-dn>
            <arg-dn>
                <token-text>Users\Manager
Template</token-text>
            </arg-dn>
        </do-set-op-template-dn>
    </actions>
</rule>
</policy>

```

佈置規則

「佈置」規則用來決定新物件要放在哪裡，以及它們在 Identity Vault 和已連接應用程式中的名稱。

如果您要在 Identity Vault 中進行物件建立，則「發行者」通道上必須要有「佈置」規則。即使您要在已連接應用程式中進行物件建立，「訂閱者」通道上也不一定要有「佈置」規則，要看目的資料儲存的本質而定。例如，由於關聯式資料庫中的列沒有位置或名稱，所以當同步化關聯式資料庫時，就不需要「佈置」規則。

範例：

- ◆ 依屬性值佈置
- ◆ 依名稱佈置

依屬性值佈置：此範例「DirXML 程序檔」規則會根據 Department 屬性的值，將「使用者」建立在特定的容器中。

```

<policy>
    <rule>
        <description>Department Engineering</description>
        <conditions>
            <and>
                <if-class-name op="equal">User</if-class-name>
                <if-op-attr mode="regex" name="Department"
op="equal">.*Engineering.*</if-op-attr>
            </and>
        </conditions>
        <actions>
            <do-set-op-dest-dn>
                <arg-dn>
                    <token-text>Eng</token-text>
                    <token-text>\</token-text>
                    <token-op-attr name="CN"/>
                </arg-dn>
            </do-set-op-dest-dn>
        </actions>
    </rule>
</policy>

```

```

        </actions>
    </rule>
    <rule>
        <description>Department HR</description>
        <conditions>
            <and>
                <if-class-name op="equal">User</if-class-
name>
                <if-op-attr mode="regex" name="Department"
op="equal">.*HR.*</if-op-attr>
            </and>
        </conditions>
        <actions>
            <do-set-op-dest-dn>
                <arg-dn>
                    <token-text>HR</token-text>
                    <token-text>\</token-text>
                    <token-op-attr name="CN"/>
                </arg-dn>
            </do-set-op-dest-dn>
        </actions>
    </rule>
</policy>

```

此範例「DirXML 程序檔」規則會根據輸入文件中的 `src-dn` 來決定「使用者」或「組織單位」的位置。

```

<policy>
    <rule>
        <description>PublisherPlacementRule</description>
        <conditions>
            <or>
                <if-class-name op="equal">User</if-class-
name>
                <if-class-name op="equal">Organizational
Unit</if-class-name>
            </or>
            <or>
                <if-src-dn op="in-subtree">o=people,
o=novell</if-src-dn>
            </or>
        </conditions>
        <actions>
            <do-set-op-dest-dn>
                <arg-dn>
                    <token-text>People</token-text>
                    <token-text>\</token-text>
                    <token-unmatched-src-dn convert="true"/>
                </arg-dn>
            </do-set-op-dest-dn>
        </actions>
    </rule>

```



```
</policy>
```

依名稱佈置：此範例「DirXML 程序檔」規則會根據使用者姓氏的第一個字母，將使用者建立在特定的容器中。使用者的姓氏以 A 到 I 開頭者，會放在容器 Users1 中，而姓氏開頭為 J 到 R 的使用者，則會放在 Users2 中，S 到 Z 則放在 Users3 中。

```
<policy>
  <rule>
    <description>Surname - A to I in Users1</description>
    <conditions>
      <and>
        <if-class-name op="equal">User</if-
class-name>
        <if-op-attr mode="regex" name="Surname"
op="equal">[A-I].*</if-op-attr>
      </and>
    </conditions>
    <actions>
      <do-set-op-dest-dn>
        <arg-dn>
          <token-text>Users1</token-text>
          <token-text>\</token-text>
          <token-op-attr name="CN"/>
        </arg-dn>
      </do-set-op-dest-dn>
    </actions>
  </rule>
  <rule>
    <description>Surname - J to R in Users2</description>
    <conditions>
      <and>
        <if-class-name op="equal">User</if-class-
name>
        <if-op-attr mode="regex" name="Surname"
op="equal">[J-R].*</if-op-attr>
      </and>
    </conditions>
    <actions>
      <do-set-op-dest-dn>
        <arg-dn>
          <token-text>Users2</token-text>
          <token-text>\</token-text>
          <token-op-attr name="CN"/>
        </arg-dn>
      </do-set-op-dest-dn>
    </actions>
  </rule>
  <rule>
    <description>Surname - S to Z in Users3</description>
    <conditions>
      <and>
        <if-class-name op="equal">User</if-class-
name>
```

```

                <if-op-attr mode="regex" name="Surname"
op="equal">[S-Z].*</if-op-attr>
                </and>
            </conditions>
            <actions>
                <do-set-op-dest-dn>
                    <arg-dn>
                        <token-text>Users3</token-text>
                        <token-text>\</token-text>
                        <token-op-attr name="CN"/>
                    </arg-dn>
                </do-set-op-dest-dn>
            </actions>
        </rule>
</policy>

```

指令轉換規則

「指令轉換」規則會以替代或加入指令的方式，改變 Identity Manager 正在傳送至目的資料儲存的指令。「指令轉換」規則中替代指令的一個例子是：攔截「刪除」指令，並以「修改」、「移動」或「停用」指令來取代。而「指令轉換」規則中加入指令的常見例子，則是根據「新增」指令的屬性值來建立「修改」指令。

最籠統的說法是，「指令轉換」規則是用來將 Identity Manager 執行的指令，改變為提交至 Metadirectory 引擎之事件的預設處理結果。

將不完全符合所有其他規則描述的規則包含在內，也是常見的作法。

範例：

- ◆ 將「刪除」轉換為「修改」和「移動」
- ◆ 建立其他操作
- ◆ 設定密碼過期時間

將「刪除」轉換為「修改」：此「DirXML 程序檔」規則會將 Login Disabled 屬性的「刪除」操作轉換為「修改」操作。

```

<policy>
    <rule>
        <description>Convert User Delete to Modify</description>
        <conditions>
            <and>
                <if-operation op="equal">delete</if-
operation>
                <if-class-name op="equal">User</if-class-name>
            </and>
        </conditions>
        <actions>
            <do-set-dest-attr-value name="Login Disabled">
                <arg-value type="state">
                    <token-text>>true</token-text>
                </arg-value>
            </do-set-dest-attr-value>

```

```

        <do-veto/>
    </actions>
</rule>
</policy>

```

建立其他操作：此「DirXML 程序檔」規則會判斷使用者的目的容器是否已存在。如果容器不存在，規則便會建立「新增」操作，以便建立「容器」物件。

```

<policy>
  <rule>
    <description>Check if destination container already
exists</description>
    <conditions>
      <and>
        <if-operation op="equal">add</if-operation>
      </and>
    </conditions>
    <actions>
      <do-set-local-variable name="target-container">
        <arg-string>
          <token-dest-dn length="-2"/>
        </arg-string>
      </do-set-local-variable>
      <do-set-local-variable name="does-target-exist">
        <arg-string>
          <token-dest-attr class-
name="OrganizationalUnit" name="objectclass">
            <arg-dn>
              <token-local-variable
name="target-container"/>
            </arg-dn>
          </token-dest-attr>
        </arg-string>
      </do-set-local-variable>
    </actions>
  </rule>
  <rule>
    <description>Create the target container if necessary</
description>
    <conditions>
      <and>
        <if-local-variable name="does-target-exist"
op="available"/>
        <if-local-variable name="does-target-exist"
op="equal"/>
      </and>
    </conditions>
    <actions>
      <do-add-dest-object class-name="organizationalUnit"
direct="true">
        <arg-dn>
          <token-local-variable name="target-
container"/>

```

```

        </arg-dn>
    </do-add-dest-object>
    <do-add-dest-attr-value direct="true" name="ou">
        <arg-dn>
            <token-local-variable name="target-
container"/>
        </arg-dn>
        <arg-value type="string">
            <token-parse-dn dest-dn-format="dot"
length="1" src-dn-format="dest-dn" start="-1">
                <token-local-variable
name="target-container"/>
            </token-parse-dn>
        </arg-value>
    </do-add-dest-attr-value>
</actions>
</rule>
</policy>

```

設定密碼過期時間：此「DirXML 程序檔」規則會修改 eDirectory 使用者的 Password Expiration Time 屬性。

```

<?xml version="1.0" encoding="UTF-8"?>
<policy xmlns:jssystem="http://www.novell.com/nxsl/java/
java.lang.System">
    <rule>
        <description>Set password expiration time for a given
interval from current day</description>
        <conditions>
            <and>
                <if-operation op="equal">modify-password</if-
operation>
            </and>
        </conditions>
        <actions>
            <do-set-local-variable name="interval">
                <arg-string>
                    <token-text>30</token-text>
                </arg-string>
            </do-set-local-variable>
            <do-set-dest-attr-value class-name="User"
name="Password Expiration Time" when="after">
                <arg-association>
                    <token-association/>
                </arg-association>
                <arg-value type="string">
                    <token-
xpath expression="round(jssystem:currentTimeMillis() div 1000 +
(86400*$interval))"/>
                </arg-value>
            </do-set-dest-attr-value>
        </actions>
    </rule>

```

</policy>

綱要映射規則

「綱要映射」規則會保留 Identity Vault 和已連接系統之間的綱要映射定義。

Identity Vault 綱要會從 eDirectory 讀取。已連接系統的 Identity Manager 驅動程式會提供已連接應用程式的綱要。在識別出兩個綱要後，Identity Vault 和目標應用程式之間便會建立簡易映射。

在 Identity Manager 驅動程式組態中定義了「綱要映射」規則後，便可映射對應的資料。

請務必注意以下事項：

- ◆ 相同的規則會套用於雙向。
- ◆ 在 Metadirectory 引擎和應用程式 Shim 之間的任一通道上，按任一方向傳遞的所有文件，都會透過「綱要映射」規則傳遞。

如需管理資訊，請參閱第 7 章「管理綱要映射規則」，第 389 頁。

範例：

- ◆ 基本「綱要映射」規則
- ◆ 自定「綱要映射」規則

基本綱要映射規則：此範例「DirXML 程序檔」規則會顯示基本「綱要映射」規則的原始 XML 來源。不過，當您透過 Designer for Identity Manager 編輯規則時，預設「綱要映射」編輯器便可讓規則以圖形化的方式顯示和編輯。

```
<?xml version="1.0" encoding="UTF-8"?><attr-name-map>
  <class-name>
    <app-name>WorkOrder</app-name>
    <nds-name>DirXML-nwoWorkOrder</nds-name>
  </class-name>
  <class-name>
    <app-name>PbxSite</app-name>
    <nds-name>DirXML-pbxSite</nds-name>
  </class-name>
  <attr-name class-name="DirXML-pbxSite">
    <app-name>PBXName</app-name>
    <nds-name>DirXML-pbxName</nds-name>
  </attr-name>
  <attr-name class-name="DirXML-pbxSite">
    <app-name>TelephoneNumber</app-name>
    <nds-name>Telephone Number</nds-name>
  </attr-name>
  <attr-name class-name="DirXML-pbxSite">
    <app-name>LoginName</app-name>
    <nds-name>DirXML-pbxLoginName</nds-name>
  </attr-name>
  <attr-name class-name="DirXML-pbxSite">
    <app-name>Password</app-name>
    <nds-name>DirXML-pbxPassword</nds-name>
  </attr-name>
</attr-name-map>
```

```

    </attr-name>
    <attr-name class-name="DirXML-pbxSite">
        <app-name>Nodes</app-name>
        <nds-name>DirXML-pbxNodesNew</nds-name>
    </attr-name>
</attr-name-map>

```

自定綱要映射規則：此範例「DirXML 程序檔」規則會使用「DirXML 程序檔」執行自定「綱要映射」。

```

<?xml version="1.0" encoding="UTF-8"?><policy>
  <rule>
    <!--
    The Schema Mapping Policy can only handle one-to-one
mappings.
    That Mapping Policy maps StudentPersonal addresses.
    This rule maps StaffPersonal addresses.
    -->
    <description>Publisher Staff Address Mappings</
description>
    <conditions>
      <and>
        <if-local-variable name="fromNds"
op="equal">>false</if-local-variable>
        <if-xpath op="true">@original-class-name =
'StaffPersonal'</if-xpath>
      </and>
    </conditions>
    <actions>
      <do-rename-op-attr dest-name="SA" src-name="Address/
Street/Line1"/>
      <do-rename-op-attr dest-name="Postal Office Box"
src-name="Address/Street/Line2"/>
      <do-rename-op-attr dest-name="Physical Delivery
Office Name" src-name="Address/City"/>
      <do-rename-op-attr dest-name="S" src-name="Address/
StatePr"/>
      <do-rename-op-attr dest-name="Postal Code" src-
name="Address/PostalCode"/>
    </actions>
  </rule>
  <rule>
    <description>Subscriber Staff Address Mappings</
description>
    <!--
    The Schema Mapping Policy has already mapped addresses to
StudentPersonal.
    This rule maps StudentPersonal to StaffPersonal.
    -->
    <conditions>
      <and>
        <if-local-variable name="fromNds"
op="equal">>true</if-local-variable>

```

```

        <if-op-attr name="DirXML-sifIsStaff"
op="equal">true</if-op-attr>
        </and>
    </conditions>
    <actions>
        <do-rename-op-attr dest-name="Address/Street/Line1"
src-name="StudentAddress/Address/Street/Line1"/>
        <do-rename-op-attr dest-name="Address/Street/Line2"
src-name="StudentAddress/Address/Street/Line2"/>
        <do-rename-op-attr dest-name="Address/City" src-
name="StudentAddress/Address/City"/>
        <do-rename-op-attr dest-name="Address/StatePr" src-
name="StudentAddress/Address/StatePr"/>
        <do-rename-op-attr dest-name="Address/PostalCode"
src-name="StudentAddress/Address/PostalCode"/>
    </actions>
</rule>
</policy>

```

輸出轉換規則

「輸出轉換」規則主要處理由 Metadirectory 引擎提供的資料到應用程式 Shim 所需的資料之間，一種資料格式上的轉換。這些轉換的範例包含：

- ◆ 屬性值的格式轉換
- ◆ XML 詞彙轉換
- ◆ 「輸出轉換」規則也可讓您自定處理由 Metadirectory 引擎傳回到應用程式 Shim 的狀態訊息

Metadirectory 引擎在任一通道上提供給應用程式 Shim 的所有文件，都會透過「輸出轉換」規則傳遞。由於「輸出轉換」發生於綱要映射之後，因此所有的綱要名稱都會在應用程式名稱空間內。

範例：

- ◆ 屬性值的格式轉換
- ◆ 自定處理狀態訊息

屬性值的轉換 此範例「DirXML 程序檔」規則會將電話號碼由 (nnn) nnn-nnnn 格式，重新格式化為 nnn.nnn.nnnn 格式。您可以在「輸入轉換」規則範例中找到反向轉換。

```

<policy>
    <rule>
        <description>Reformat all telephone numbers from (nnn)
nnn-nnnn to nnn.nnn.nnnn</description>
        <conditions/>
        <actions>
            <do-reformat-op-attr name="telephoneNumber">
                <arg-value type="string">
                    <token-replace-first
regex="^\((\d\d\d)\) *(\d\d\d)-(\d\d\d\d)$" replace-with="$1.$2.$3">
                        <token-local-

```

```

variable name="current-value"/>
                                </token-replace-first>
                                </arg-value>
                                </do-reformat-op-attr>
                                </actions>
                                </rule>
</policy>

```

自定處理狀態訊息：此範例「DirXML 程序檔」規則會偵測層級不等於 success，且又包含操作資料內之子代 password-publish-status 元素的狀態文件，然後再使用 DoSendEmailFromTemplate 動作產生電子郵件訊息。

```

<?xml version="1.0" encoding="UTF-8"?>
  <policy>
    <description>Email notifications for failed password
publications</description>
    <rule>
      <description>Send e-mail for a failed publish
password operation</description>
      <conditions>
        <and>
          <if-global-variable
mode="nocase" name="notify-user-on-password-dist-failure"
op="equal">true</if-global-variable>
          <if-operation
op="equal">status</if-operation>
          <if-xpath
op="true">self::status[@level != 'success']/operation-data/password-
publish-status</if-xpath>
        </and>
      </conditions>
      <actions>
        <!-- generate email notification -->
        <do-send-email-from-template notification-
dn="\cn=security\cn=Default Notification Collection" template-
dn="\cn=security\cn=Default Notification Collection\cn>Password Sync
Fail">
          <arg-string name="UserFullName">
            <token-src-attr name="Full Name">
              <arg-association>
                <token-xpath
expression="self::status/operation-data/password-publish-status/
association"/>
              </arg-association>
            </token-src-attr>
          </arg-string>
          <arg-string name="UserGivenName">
            <token-src-attr name="Given Name">
              <arg-association>
                <token-xpath
expression="self::status/operation-data/password-publish-status/
association"/>
              </arg-association>
            </arg-string>
          </arg-association>
        </do-send-email-from-template>
      </actions>
    </rule>
  </policy>

```



```

        </token-src-attr>
    </arg-string>
    <arg-string name="UserLastName">
        <token-src-attr name="Surname">
            <arg-association>
                <token-xpath
expression="self::status/operation-data/password-publish-status/
association"/>
            </arg-association>
        </token-src-attr>
    </arg-string>
    <arg-string name="ConnectedSystemName">
        <token-global-variable
name="ConnectedSystemName"/>
    </arg-string>
    <arg-string name="to">
        <token-src-attr name="Internet Email
Address">
            <arg-association>
                <token-xpath
expression="self::status/operation-data/password-publish-status/
association"/>
            </arg-association>
        </token-src-attr>
    </arg-string>
    <arg-string name="FailureReason">
        <token-text/>
        <token-xpath
expression="self::status/child::text()"/>
    </arg-string>
    </do-send-email-from-template>
</actions>
</rule>
</policy>

```

輸入轉換規則

「輸入轉換」規則主要處理由應用程式 Shim 提供給 Metadirectory 引擎所需的資料之間，一種資料格式上的轉換。這些轉換的範例包括：

- ◆ 屬性值的格式轉換
- ◆ XML 詞彙轉換
- ◆ 驅動程式活動訊號
- ◆ 「輸入轉換」規則也可讓您自定處理由應用程式 Shim 傳回到 Metadirectory 引擎的狀態訊息。

應用程式 Shim 在任一通道上提供給 Metadirectory 引擎的所有文件，都會透過「輸入轉換」規則傳遞。

範例：

- ◆ 屬性值的格式轉換

- ◆ 驅動程式活動訊號

屬性值的格式轉換：此範例「DirXML 程序檔」規則會將電話號碼由 nnn.nnn.nnnn 格式，重新格式化為 (nnn) nnn-nnnn 格式。您可以在「輸出轉換」規則範例中找到反向轉換。

```
<policy>
  <rule>
    <description>Reformat all telephone numbers from
    nnn.nnn.nnnn to (nnn) nnn-nnnn</description>
    <conditions/>
    <actions>
      <do-reformat-op-attr name="telephoneNumber">
        <arg-value type="string">
          <token-replace-first
            regex="^(\\d\\d\\d)\\. (\\d\\d\\d)\\. (\\d\\d\\d\\d)$" replace-with="(\\$1) \\$2-\\$3">
<token-local-variable name="current-value"/>
          </token-replace-first>
        </arg-value>
      </do-reformat-op-attr>
    </actions>
  </rule>
</policy>
```

驅動程式活動訊號：此「DirXML 程序檔」規則會建立狀態活動訊號事件。驅動程式的活動訊號功能用於在每個活動訊號間隔之間傳送成功訊息 (HEARTBEAT: \$driver)。該訊息可由 Novell Audit 監看。Identity Manager 必須支援活動訊號，且必須在驅動程式組態頁面上啟用活動訊號。

```
<?xml version="1.0" encoding="UTF-8" ?>
<policy>
  <rule>
    <description>Heartbeat Rule, v1.01, 040126, by Holger Dopp</
description>
    <conditions>
      <and>
        <if-operation op="equal">status</if-operation>
        <if-xpath op="true">@type="heartbeat"</if-
xpath>
      </and>
    </conditions>
    <actions>
      <do-set-xml-attr expression="." name="text1">
        <arg-string>
          <token-global-variable
name="dirxml.auto.driverdn" />
        </arg-string>
      </do-set-xml-attr>
      <do-set-xml-attr expression="." name="text2">
        <arg-string>
          <token-text>HEARTBEAT</token-text>
        </arg-string>
      </do-set-xml-attr>
```

```
</actions>
</rule>
</policy>
```

1.2.2 定義規則

所有規則會以兩種方式之一定義：

- ◆ 使用「規則產生器」介面產生「DirXML 程序檔」。現有的非 XSLT 規則會在輸入時自動轉換為「DirXML 程序檔」。
- ◆ 使用 XSLT 樣式表。

「綱要映射」規則也可以使用綱要映射表格加以定義（通常也是如此）。

規則產生器和 DirXML 程序檔

「規則產生器」介面用於定義您可能會實作的大部份規則。「規則產生器」介面使用圖形化環境，讓您可以輕鬆地定義並管理規則。

在「規則產生器」中，建立規則的基本功能是由稱為「DirXML 程序檔」的自定指令碼語言所提供。

「DirXML 程序檔」包含各種您可以測試的條件，用於執行的動作，以及要新增至規則的動態值。這些選項中的每一項都會透過智慧型的下拉式清單呈現，並且只提供您每一處的有效選項，以及常見值的快速連結。

「規則產生器」則可讓您不需要直接使用「DirXML 程序檔」。

如需「規則產生器」的相關資訊，請參閱第 2 章「[搭配 Designer 使用規則產生器定義規則](#)」，第 33 頁和第 3 章「[使用 iManager 內的規則產生器定義規則](#)」，第 197 頁。如需「DirXML 程序檔」的相關資訊，請參閱「[DirXML 程序檔](#)」，第 11 頁。

提示：就使用「規則產生器」而言雖然沒有必要，但您還是可以到「[DirXML 驅動程式開發人員套件文件 \(http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/ref/index.html\)](http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/ref/index.html)」網站取得完整的「DirXML 程序檔」參考資訊。

XSLT 樣式表

若要定義更多的複雜規則，可使用 XSLT 樣式表將某 XML 文件直接轉換為包含所需變更的另一個 XML 文件。

樣式表提供您大幅度的彈性。若轉換的動作不符合「規則產生器」中使用規則建立的可用預先定義條件和動作，此時便可使用樣式表。

若要建立 XSLT 樣式表，您便需要徹底瞭解 XSLT、nds.dtd，以及進出於 Metadirectory 引擎的指令和事件。如需詳細的 nds.dtd 參考，請參閱「[NDS DTD 參考 \(http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/ref/ndsdttd/index.html\)](http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/ref/ndsdttd/index.html)」

如需 XSLT 樣式表的相關資訊，請參閱第 5 章「[使用 XSLT 樣式表定義規則](#)」，第 351 頁。

可下載的 **Identity Manager** 規則

Novell 已提供可以下載並用於您環境中的範例規則。Novell 的支援網站 (<http://support.novell.com/filefinder/20607/index.html>) 中有這些規則可供使用。下載檔案並加以擷取。How_To_Install.rtf 檔包含了安裝指示。

若要使用 Designer 輸入檔案，請參閱「從 XML 檔案輸入規則」，第 54 頁。若要使用 iManager 輸入檔案，請參閱「從 XML 檔案輸入規則」，第 208 頁。

1.3 過濾器

過濾器會指定 Metadirectory 引擎針對其處理事件的物件類別和屬性，並指定如何處理對於那些類別和屬性的變更。

過濾器只在物件的基本類別符合其所指定的其中一項類別時，傳遞發生於該物件上的事件。過濾器不會在物件是其所指定之類別的從屬類別時，傳遞發生於該物件上的事件，除非也指定了從屬類別。每個通道都有個別的過濾器設定，可讓您控制各類別和屬性的同步化方向與授權資料來源。

附註：在 eDirectory 中，基本類別就是用於建立項目的物件類別。您必須在過濾器中指定該類別，而不是基本類別所承襲的來源 Super 類別，或其他屬性可能的來源輔助類別。

例如，如果具有 Surname 和 Given Name 屬性的「使用者」類別設為要在過濾器中進行同步化，Metadirectory 引擎便會將任何變更傳遞至那些屬性。不過，如果該項目的 Telephone Number 屬性遭到修改，則 Metadirectory 引擎會因為 Telephone Number 屬性不在過濾器中而捨棄此事件。

過濾器的組態必須設定為包含以下項目：

- ◆ 要同步化的屬性
- ◆ 不要同步化，但要用於觸發規則以執行工作的屬性

如需定義過濾器的相關資訊，請參閱第 6 章「管理過濾器」，第 365 頁。

搭配 Designer 使用規則產生器定義規則

「規則產生器」是一種完整的圖形化介面，可用於建立並管理定義已連接系統之間資料交換的規則。

這幾節會提供下列有關規則和如何使用「規則產生器」的資訊：

- ◆ 「規則」，第 33 頁
- ◆ 「Designer 中的規則產生器任務」，第 34 頁

這幾節也包含下列的詳細參考章節：

- ◆ 「一般運算式」，第 112 頁
- ◆ 「XPath 1.0 運算式」，第 113 頁
- ◆ 「條件」，第 114 頁
- ◆ 「動作」，第 130 頁
- ◆ 「名詞記號」，第 173 頁
- ◆ 「動詞記號」，第 186 頁

2.1 規則

瞭解規則的元件非常重要，這是瞭解規則如何運作的一部份。

- ◆ 規則 (Policy) 是由規則 (Rule) 組成。
- ◆ 規則 (Rule) 則是由一組條件 (請參閱「條件」，第 114 頁) 組成，必須先符合這些條件，才會發生定義的動作 (請參閱「動作」，第 130 頁)。
- ◆ 動作的動態引數可以含有衍生自執行時期所展開的記號。
- ◆ 記號可以分成兩種類別：名詞 (請參閱「名詞記號」，第 173 頁) 和動詞 (請參閱「動詞記號」，第 186 頁)。
 - ◆ 名詞記號會展開為衍生自目前操作、來源或目的資料儲存，或是某外部來源的值。
 - ◆ 動詞記號則會修改從屬於該記號之其他記號的串連結果。
- ◆ 一般運算式 (請參閱「一般運算式」，第 112 頁) 和 XPath 1.0 運算式 (請參閱「XPath 1.0 運算式」，第 113 頁) 常用於規則 (Rule) 中，以建立規則 (Policy) 的所需結果。
- ◆ 規則 (Policy) 運作於 XDS 文件上，其主要用途為檢查並修改該文件。
- ◆ 操作就是 XDS 文件中任何為輸入元素和輸出元素之子代的元素。這些元素是 Novell 之 nds.dtd 的一部份；如需相關資訊，請參閱 NDS DTD (<http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/ref/ndsdttd/index.html>)。
- ◆ 操作通常代表事件、指令或狀態。
- ◆ 規則 (Policy) 會個別套用至每項操作。當規則 (Policy) 輪流套用至每項操作時，該操作即變成目前的操作。每個規則 (Rule) 會依序套用至目前的操作。所有規則 (Rule) 會套用至目前的操作，除非先前規則 (Rule) 執行了某個使後續規則 (Rule) 不再被套用的動作。

- ◆ 規則 (Policy) 也可以從文件外部取得其他上下文位置，並造成不會反應在結果文件中的副作用。

2.2 Designer 中的規則產生器任務

本節包含在「規則產生器」中執行常見任務的指示：

- ◆ 「開啓規則產生器」，第 34 頁
- ◆ 「建立規則」，第 38 頁
- ◆ 「建立規則」，第 46 頁
- ◆ 「建立引數」，第 55 頁
- ◆ 「編輯規則」，第 65 頁
- ◆ 「使用預先定義的規則」，第 68 頁
- ◆ 「使用「規則模擬器」測試規則」，第 97 頁
- ◆ 「編輯 DirXML 程序檔」，第 106 頁

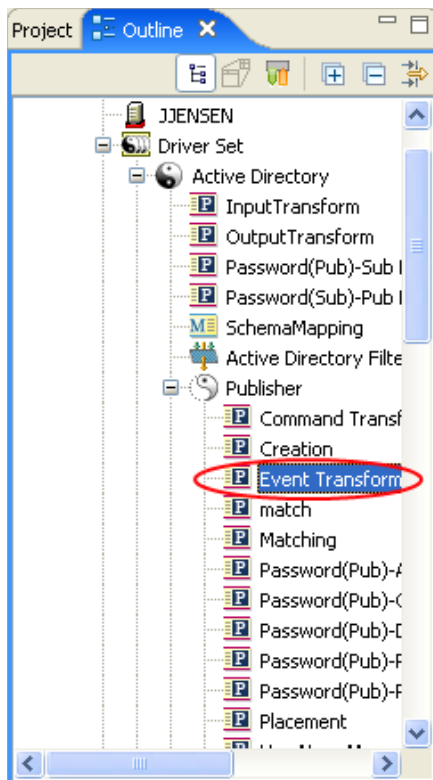
2.2.1 開啓規則產生器

「規則產生器」可以從「模型大綱」檢視窗、「規則流程」檢視窗或規則集中開啓。

模型大綱檢視窗

- 1 在 Designer 中開啓專案。
- 2 按一下「大綱」索引標籤 > 選取「顯示模型大綱」圖示。

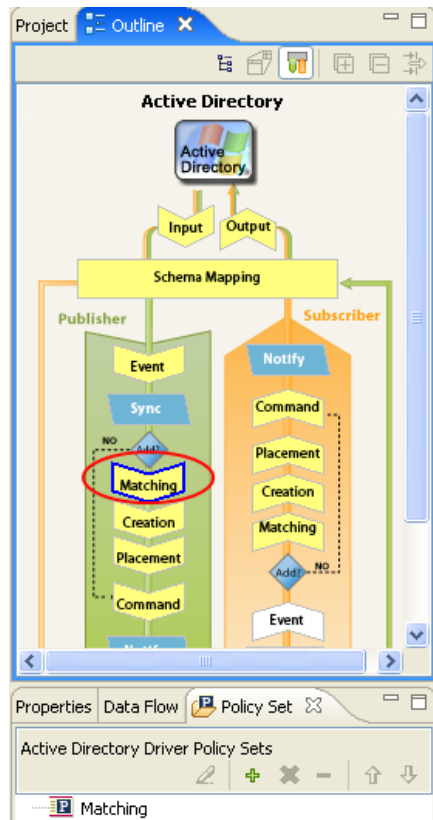
- 3 連接兩下「模型大綱」檢視窗中列出的規則，或以滑鼠右鍵按一下「編輯」。



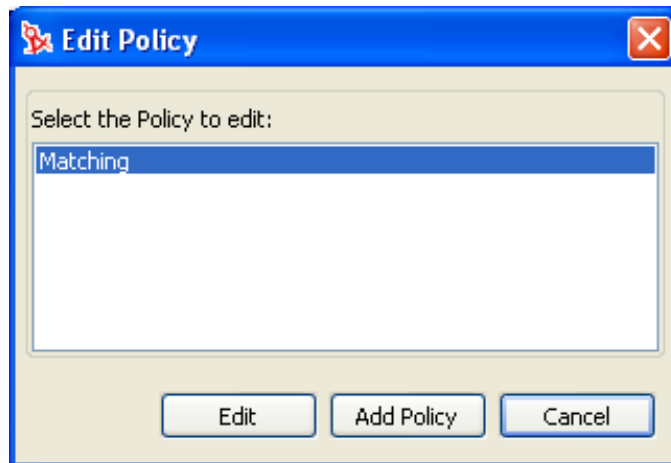
規則流程檢視窗

- 1 在 Designer 中開啓專案。
- 2 選取「大綱」索引標籤 > 選取「顯示規則流程」圖示。

- 3 以滑鼠右鍵按一下「規則流程」檢視窗中的規則（如「相符」規則），然後選取「編輯規則」。

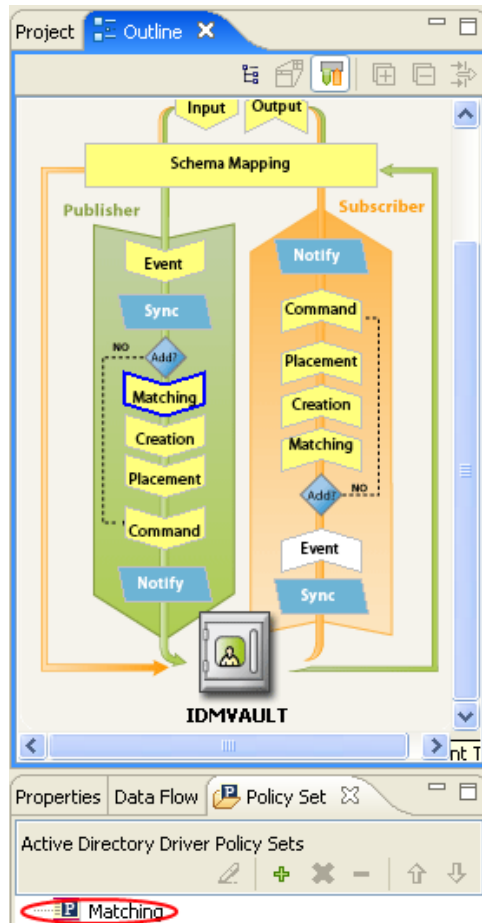


- 4 您也可以連接兩下「規則流程」中的「相符」規則。
- 5 選取規則，然後按一下「編輯」。



規則集

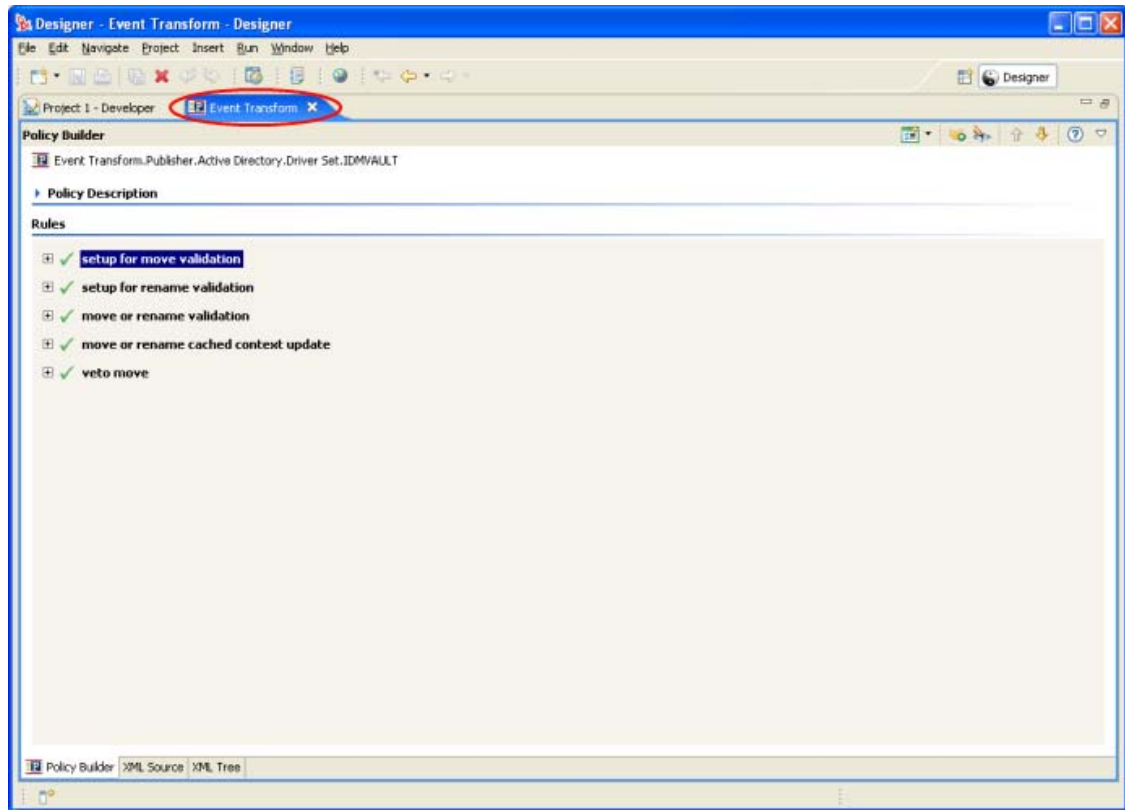
- 1 以滑鼠右鍵按一下規則集中的規則，然後按一下「編輯」。



- 2 您也可以選取規則集中的規則，然後按一下「編輯規則」圖示。

若要以不捲動的方式查看「規則產生器」視窗中的所有資訊，請連按兩下規則索引標籤，讓「規則產生器」填滿整個視窗。若要將視窗縮到最小，則再次連按兩下規則索引標籤。

特性 2-1 規則產生器的完整畫面



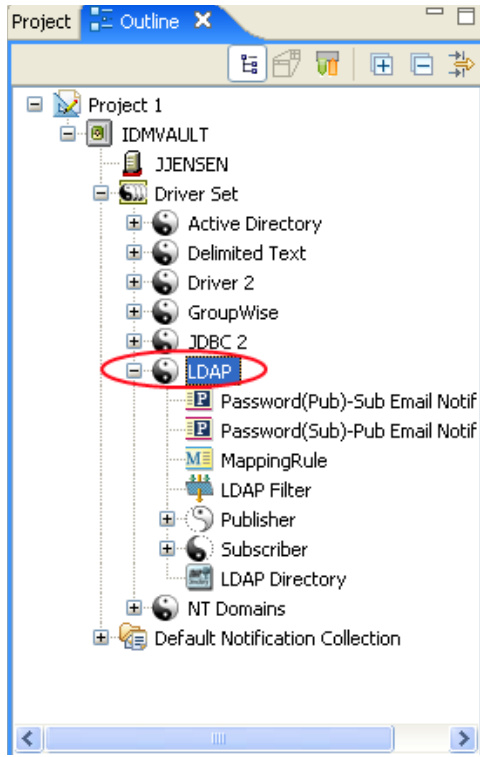
2.2.2 建立規則

規則會將資料傳送至已連接系統。規則是透過規則集建立的。

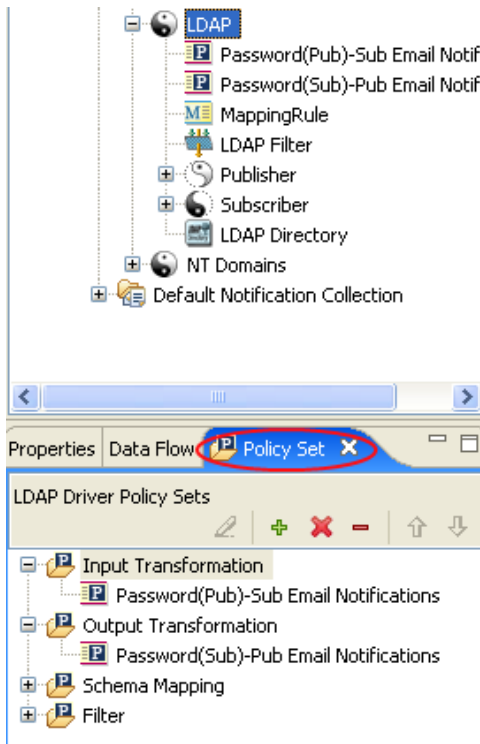
- ◆ 「存取規則集」，第 39 頁
- ◆ 「使用規則集」，第 40 頁
- ◆ 「使用「新增規則精靈」」，第 42 頁

存取規則集

- 1 從開啓專案中的「大綱」檢視窗選取驅動程式物件。

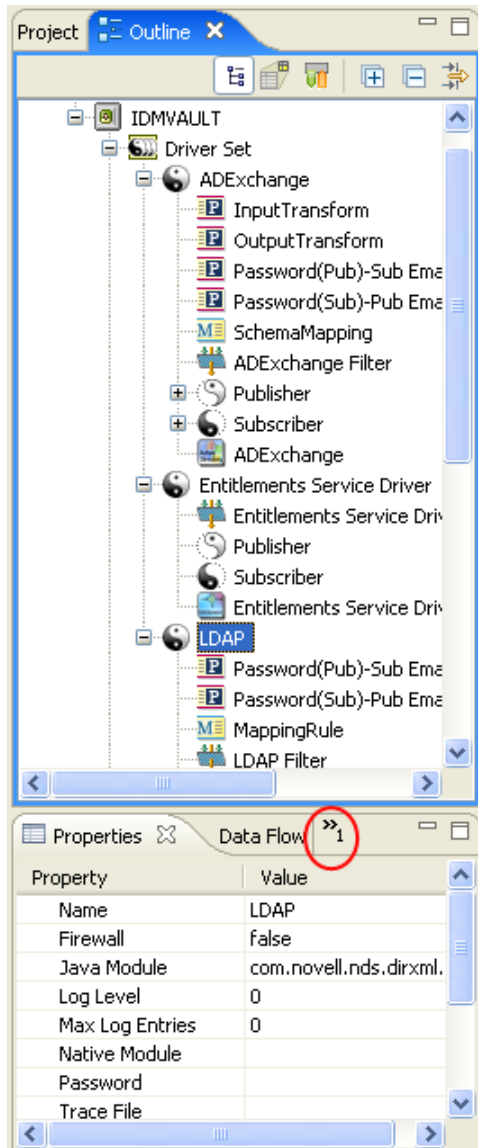


- 2 選取「規則集」索引標籤。

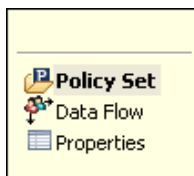


如果「規則集」索引標籤未顯示：

- 1 按一下雙箭頭。



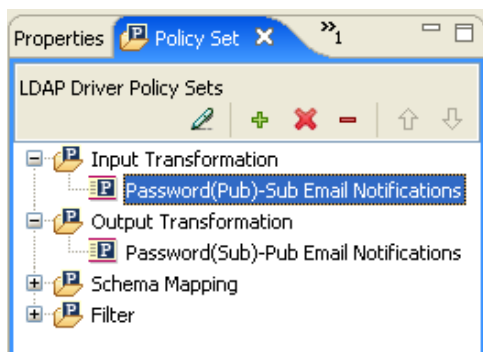
- 2 選取「規則集」。



使用規則集

規則集包含工具列和規則清單。

規則清單會顯示包含於選定規則集的所有規則。轉換時，清單中的規則是由上往下執行。工具列包含按鈕和下拉式功能表，可用於管理清單中顯示的規則，包括編輯、新增、刪除、重新命名及變更規則的處理順序。



規則集工具列

規則集會顯示規則的副本。工具列上的按鈕會隨選取的項目而啟用或停用。以下會說明各種圖示。

表格 2-1 規則集工具列

操作	說明
編輯規則	啟動「規則產生器」。
建立或新增新規則至「規則集」	啟動「新增規則精靈」。
移除並刪除選定的規則	從專案中刪除規則。
從選定規則集中移除規則，但不將其刪除	從選定規則集物件中移除規則，但不將其刪除。
在規則鏈中上移規則	將規則在處理順序中的位置上移。
在規則鏈中下移規則	將規則在處理順序中的位置下移。

鍵盤支援

您可以使用按鍵和滑鼠瀏覽規則集。受支援的按鍵列於下方。

表格 2-2 鍵盤支援

按鍵	描述
向上鍵	將選取的規則在處理順序中的位置上移。
向下鍵	將選取的規則在處理順序中的位置下移。
刪除	從專案中刪除規則。
- (減號鍵)	從選取規則集中移除規則，但不將其刪除。
+ (加號鍵)	啟動「新增規則精靈」。
Ctrl+Z	復原上一個操作。

按鍵	描述
Ctrl+Y	取消復原上一個操作。

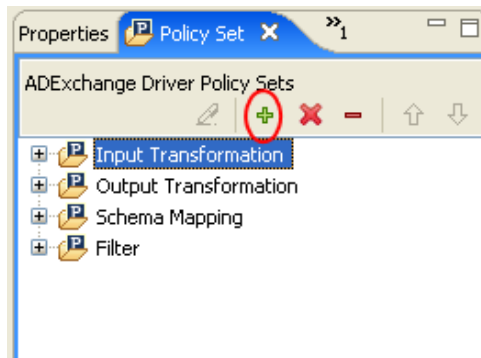
使用「新增規則精靈」

「新增規則精靈」會在您按一下工具列內的「建立或新增新規則至規則集」圖示時啟動。「新增規則精靈」可讓您執行下列動作：

- ◆ 「建立規則」，第 42 頁
- ◆ 「複製規則」，第 44 頁
- ◆ 「連結至規則」，第 45 頁

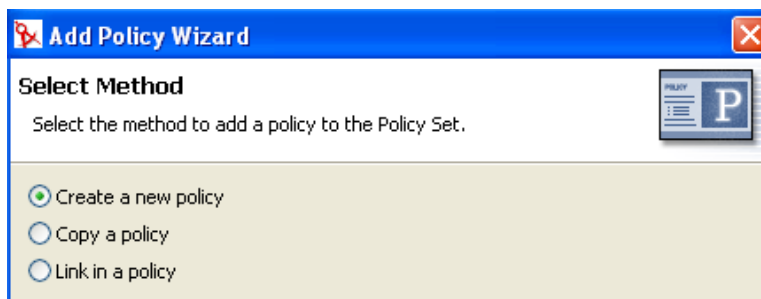
若要啟動「新增規則精靈」，請執行下列動作：

- 1 在「大綱」檢視窗中選取驅動程式。
- 2 在規則集中選取規則集項目，然後按一下工具列內的「建立或新增新規則至規則集」圖示。

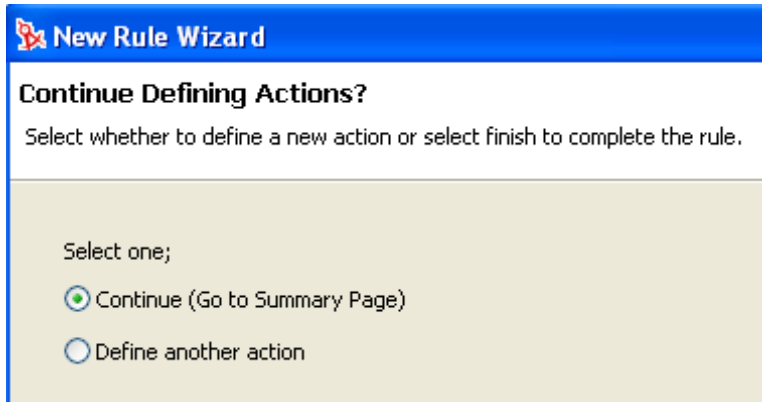


建立規則

- 1 在「新增規則精靈」中，選取「建立新規則」，然後按「下一步」。



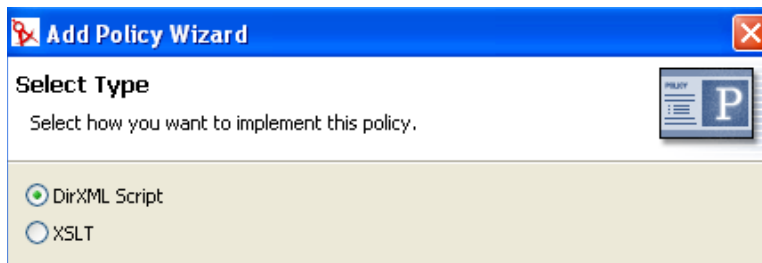
2 提供規則名稱。



3 接受預設的容器，或瀏覽至要在其中建立規則的「驅動程式」、「發行者」或「訂閱者」物件，並加以選取。

此項決定依據您要如何組織規則而定。在預設狀態下，規則會放置在啟動「新增規則精靈」時，於「大綱」索引標籤內選取的容器物件下。例如，若您在「大綱」索引標籤內移至某「發行者」物件，然後新增規則至規則集，則該規則預設會放置在 Publisher 容器下。若您要在不同的容器內建立規則，可以變更此項設定。例如，您可以在虛構的驅動程式下設定一個規則庫，將所有常用的規則放置於此驅動程式下，然後只需從其他的驅動程式參考這些規則即可。這樣一來，該規則就會是公用的規則。若您需要變更規則，僅需進行一次即可。若某規則不會被多個驅動程式重複使用，您通常可在使用該規則的驅動程式或通道之下建立該規則。

4 選取要實作的規則類型。規則類型預設為「DirXML 程序檔」。若您不希望使用「DirXML® 程序檔」，則可以選取「XSLT」或「綱要映射」。



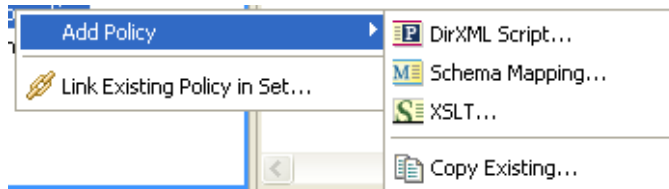
5 按一下「完成」。

若選取了「綱要映射」規則，將會提供「綱要映射」的其他選項。新規則會出現在展開的規則集中。

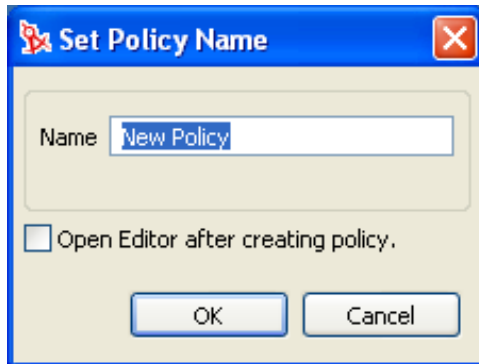
您也可以對某規則集按一下滑鼠右鍵，以新增規則。

- 1 在某規則集上按一下滑鼠右鍵（例如：「輸入轉換規則集」）。
- 2 選取「新增規則」。

- 3 選取要實作規則的方法：「DirXML 程序檔、網要映射、XSLT」或「複製現有的規則」。



- 4 為規則命名。

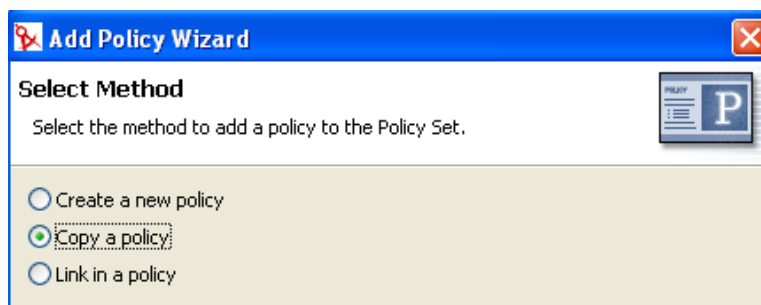


- 5 按一下「建立規則後開啓編輯器」。

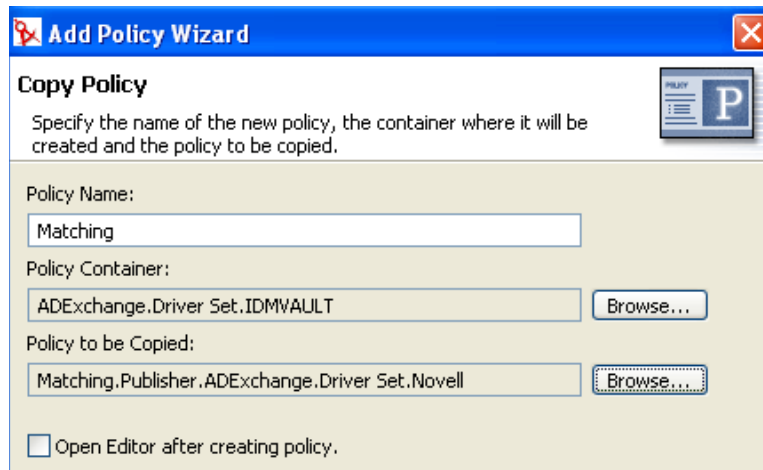
- 6 按一下「確定」。

複製規則

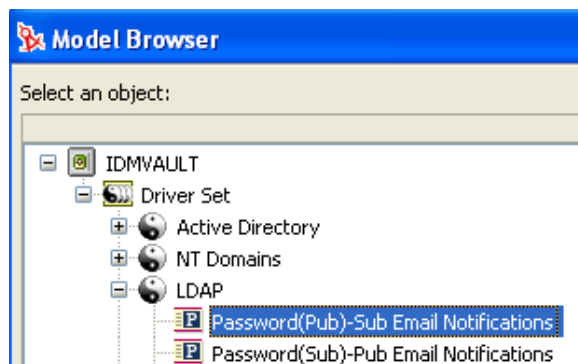
- 1 在「新增規則精靈」中，選取「複製規則」，然後按「下一步」。



2 為規則命名。



3 接受預設的容器，或瀏覽至要在其中建立規則的「驅動程式」、「發行者」或「訂閱者」物件，並加以選取。

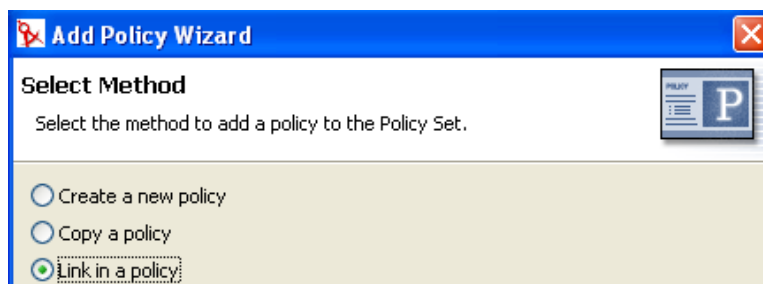


4 瀏覽至要複製的規則並加以選取，然後按一下「確定」。

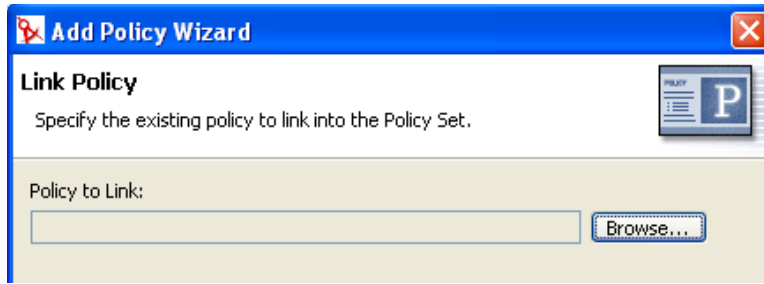
5 按一下「完成」，為選取的規則建立副本。

連結至規則

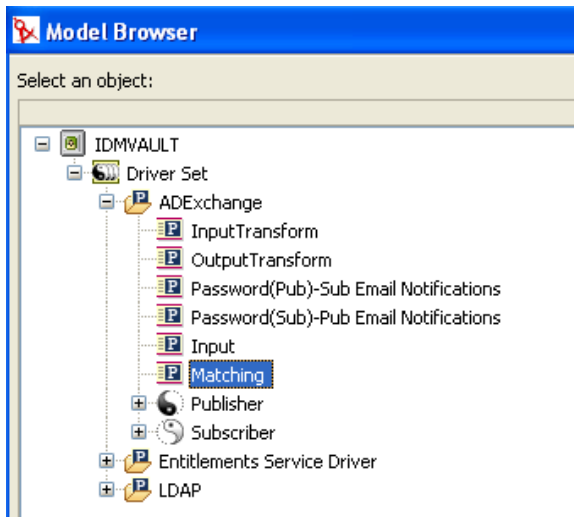
1 在「新增規則精靈」中，選取「連結規則」，然後按「下一步」。



- 按一下「瀏覽」啓動模型瀏覽器。



- 瀏覽至要連結之規則集的「規則」物件並加以選取，然後按一下「確定」。



將規則連結至規則集不會建立新的「規則」物件，而是新增一項參考至現有的規則。此參考可以是任何位於目前 Identity Vault 內的現有規則。此參考不需要包含在目前的「驅動程式」物件中，但規則類型必須是將連結之規則集的有效規則類型。例如，您不能將「綱要映射」規則連結至「輸入」規則集。

在檢視所有規則時，不允許將某規則連結至規則集。

- 按一下「完成」，以連結至選取的規則。


2.2.3 建立規則

規則 (Rule) 的定義是指定義的動作發生前必須先符合的一組條件。規則是從條件群組、條件及動作中建立的。

您可以使用下列四種不同的方式建立規則：

- ◆ 「建立新規則」，第 47 頁
- ◆ 「使用預先定義的規則」，第 51 頁
- ◆ 「包含現有的規則」，第 52 頁
- ◆ 「從 XML 檔案輸入規則」，第 54 頁

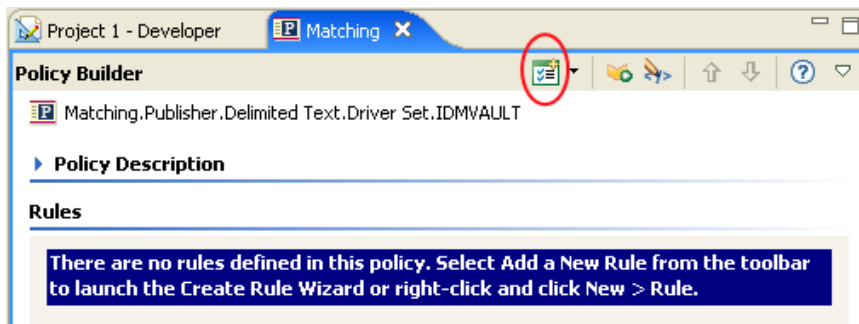
建立新規則

在建立規則時，您會建立條件群組、條件和動作。每個規則皆由條件、動作和引數所組成。如需相關資訊，請在建立每一個項目時，按一下「說明」圖示 。說明檔案包含使用之項目的定義及範例。

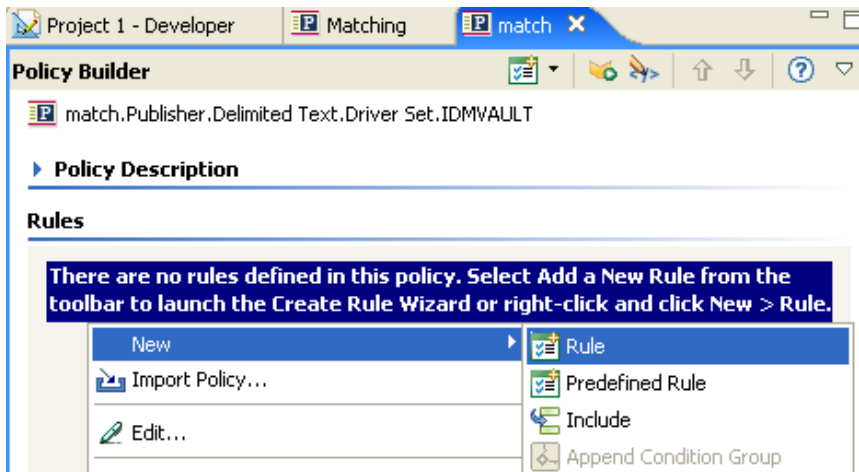
- ◆ 「建立規則」，第 47 頁
- ◆ 「建立條件式群組」，第 50 頁
- ◆ 「建立條件」，第 51 頁
- ◆ 「建立動作」，第 51 頁

建立規則

- 1 從「規則產生器」工具列中，選取「規則」。

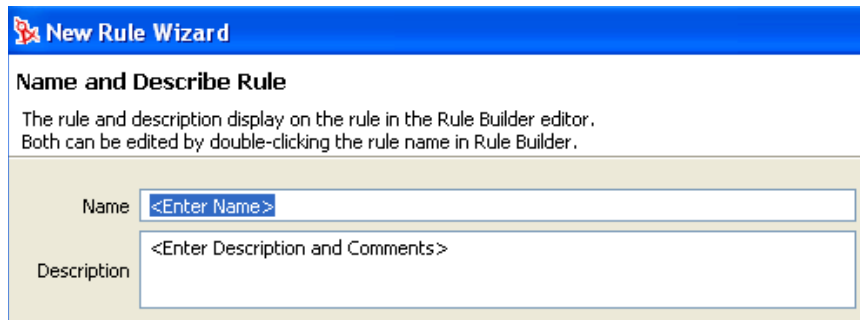


您也可以按一下滑鼠右鍵，再按一下「新增」>「規則」。



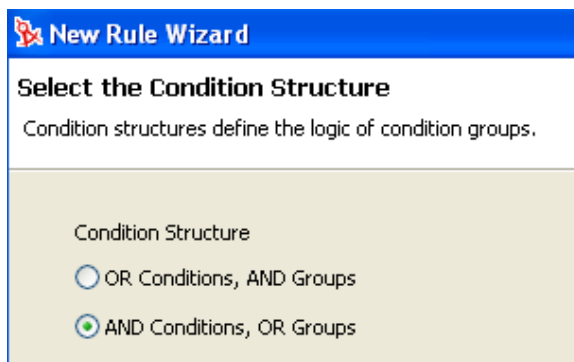
任一選項皆會啟動「建立規則精靈」。

- 2 指定規則的名稱，然後按「下一步」。



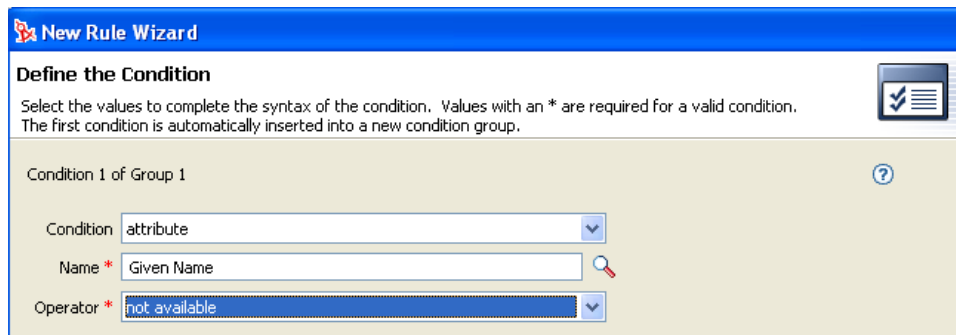
The screenshot shows the 'New Rule Wizard' dialog box with the title 'New Rule Wizard'. Below the title bar is the section 'Name and Describe Rule'. A sub-header reads: 'The rule and description display on the rule in the Rule Builder editor. Both can be edited by double-clicking the rule name in Rule Builder.' There are two input fields: 'Name' with the placeholder text '<Enter Name>' and 'Description' with the placeholder text '<Enter Description and Comments>'.

- 3 選取條件結構（「OR 條件，AND 群組」或「AND 條件，OR 群組」），然後按「下一步」。




The screenshot shows the 'New Rule Wizard' dialog box with the title 'New Rule Wizard'. Below the title bar is the section 'Select the Condition Structure'. A sub-header reads: 'Condition structures define the logic of condition groups.' There are two radio button options: 'OR Conditions, AND Groups' (which is unselected) and 'AND Conditions, OR Groups' (which is selected).

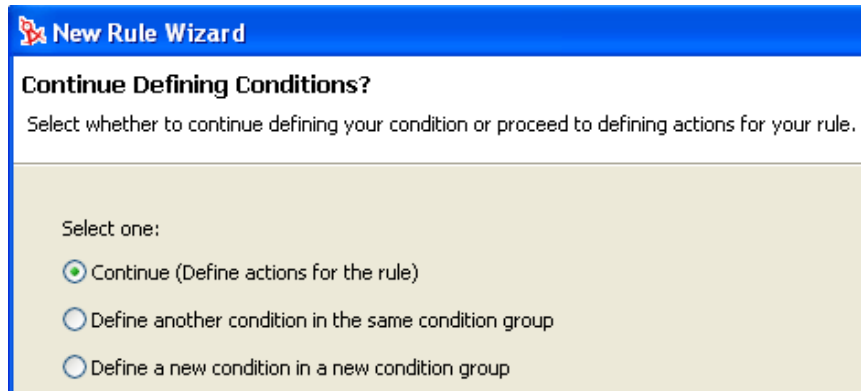
- 4 選取要使用的條件，指定適當的資訊，然後按「下一步」。



The screenshot shows the 'New Rule Wizard' dialog box with the title 'New Rule Wizard'. Below the title bar is the section 'Define the Condition'. A sub-header reads: 'Select the values to complete the syntax of the condition. Values with an * are required for a valid condition. The first condition is automatically inserted into a new condition group.' There is a 'Condition 1 of Group 1' section with a help icon (?). It contains three fields: 'Condition' with a dropdown menu showing 'attribute', 'Name *' with a text input field containing 'Given Name' and a magnifying glass icon, and 'Operator *' with a dropdown menu showing 'not available'.

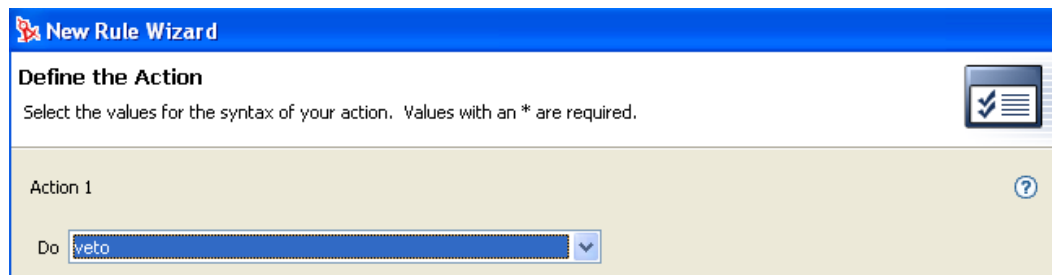
如需您能夠建立之每個條件的相關資訊，請按一下「說明」圖示 。

- 5 您可以在此時定義其他條件或條件群組。此例中只有一個條件。選取「繼續」，然後按「下一步」。




The screenshot shows the 'New Rule Wizard' dialog box with the title 'New Rule Wizard'. Below the title bar, the section is titled 'Continue Defining Conditions?'. The instruction reads: 'Select whether to continue defining your condition or proceed to defining actions for your rule.' Below this, there is a 'Select one:' label followed by three radio button options: 'Continue (Define actions for the rule)' (which is selected), 'Define another condition in the same condition group', and 'Define a new condition in a new condition group'.

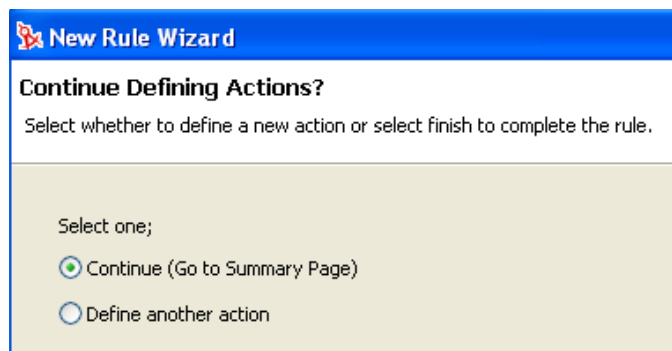
- 6 選取要採取的動作，然後按「下一步」。



The screenshot shows the 'New Rule Wizard' dialog box with the title 'New Rule Wizard'. Below the title bar, the section is titled 'Define the Action'. The instruction reads: 'Select the values for the syntax of your action. Values with an * are required.' To the right of the instruction is a 'Help' icon (a document with a checkmark). Below the instruction, there is a label 'Action 1' with a 'Help' icon to its right. Underneath, there is a dropdown menu labeled 'Do' with the value 'veto' selected.

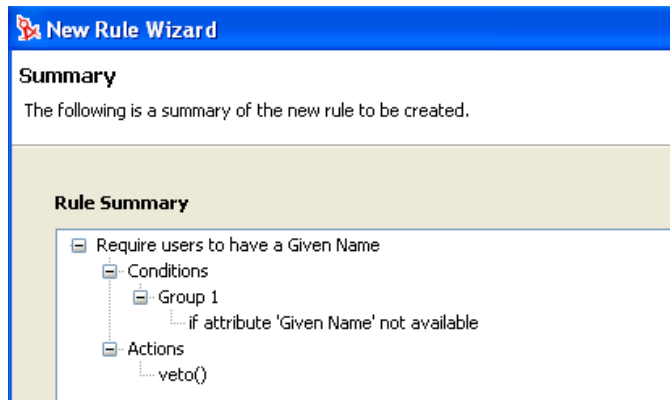
如需您能夠建立之每個動作的相關資訊，請按一下「說明」圖示 。

- 7 您可以在此時定義其他動作。此例中只有一個動作。選取「繼續」，然後按「下一步」。

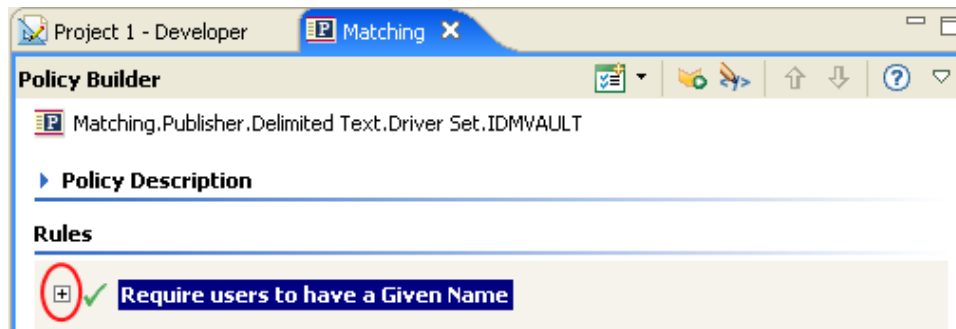


The screenshot shows the 'New Rule Wizard' dialog box with the title 'New Rule Wizard'. Below the title bar, the section is titled 'Continue Defining Actions?'. The instruction reads: 'Select whether to define a new action or select finish to complete the rule.' Below this, there is a 'Select one;' label followed by two radio button options: 'Continue (Go to Summary Page)' (which is selected) and 'Define another action'.

8 建立的規則會顯示在摘要頁面中。按一下「完成」，即可完成規則的建立程序。

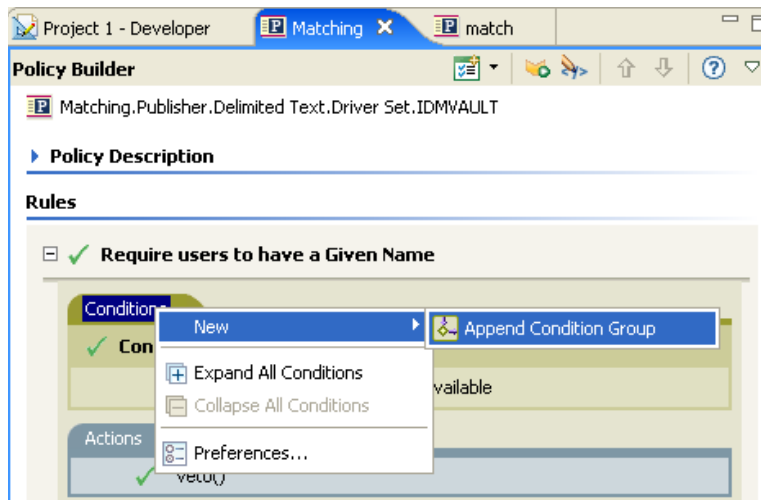


您可以按一下加號或減號，以展開或折疊規則的檢視窗。



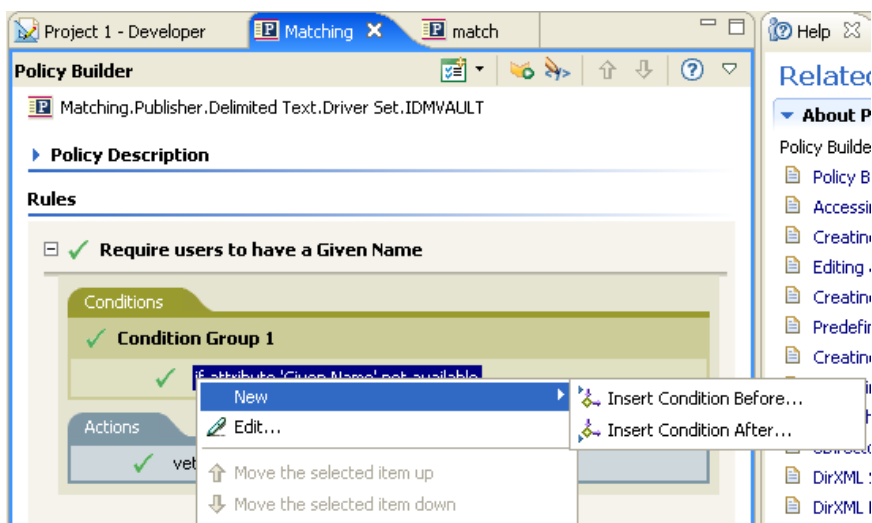
建立條件式群組

1 在「條件」索引標籤上按一下滑鼠右鍵，或在「條件式群組」的名稱上按一下滑鼠右鍵，然後按一下「新增 > 附加條件群組」。



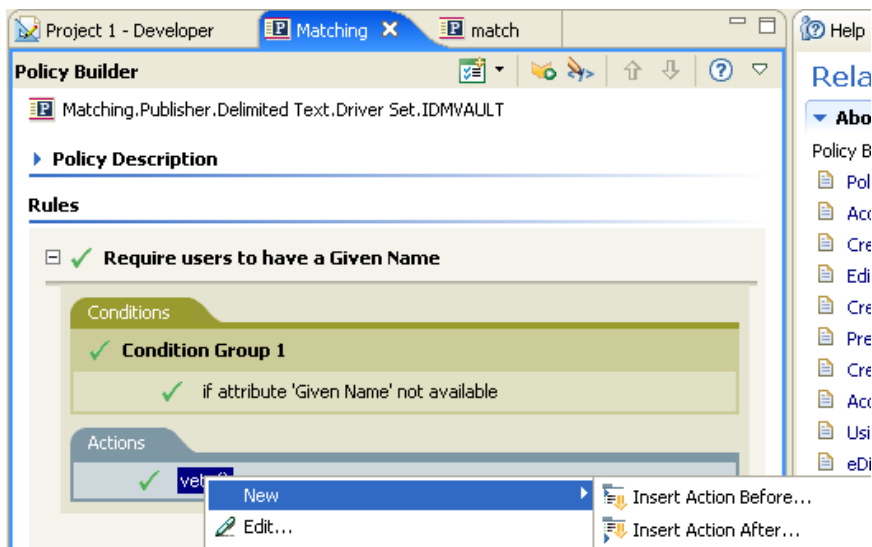
建立條件

- 1 在條件上按一下滑鼠右鍵，然後按一下「新增 > 在之前插入條件」，或「在之後插入條件」。



建立動作

- 1 在動作上按一下滑鼠右鍵，然後按一下「新增 > 在之前插入動作」，或「在之後插入動作」。

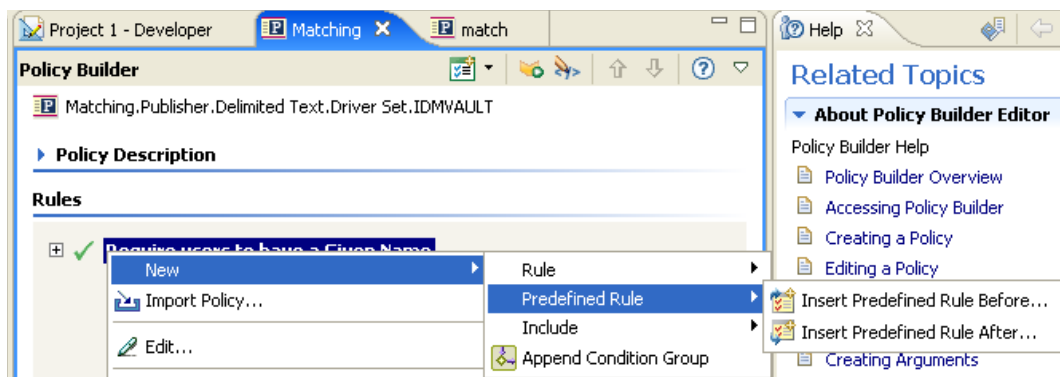


使用預先定義的規則

Designer 包含預先定義的規則清單。您可以輸入及使用這些規則，也可以建立自己的規則。

- 1 在「規則產生器」內按一下滑鼠右鍵，並選取「新增 > 預先定義的規則 > 在之前插入預先定義的規則」或「在之後插入預先定義的規則」。

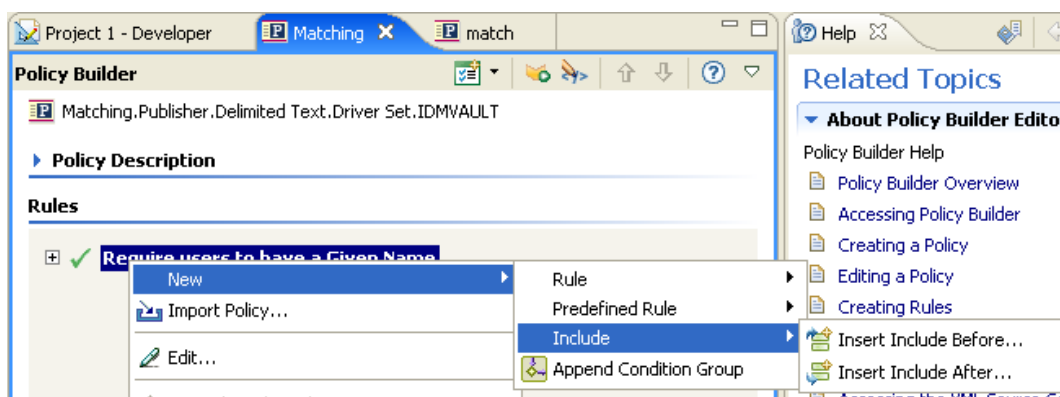
如需相關資訊，請參閱「使用預先定義的規則」，第 68 頁。



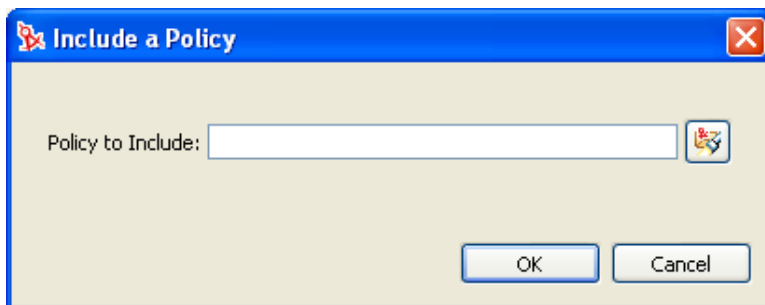
包含現有的規則

Designer 可讓您包含來自另一個規則 (Policy) 的規則 (Rule)。

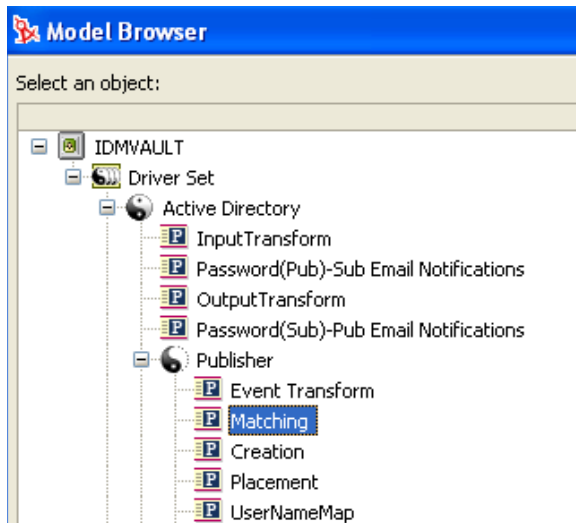
- 1 在「規則產生器」內按一下滑鼠右鍵，然後按一下「新增 > 包含 > 在之前插入包含項目」或「在之後插入包含項目」。



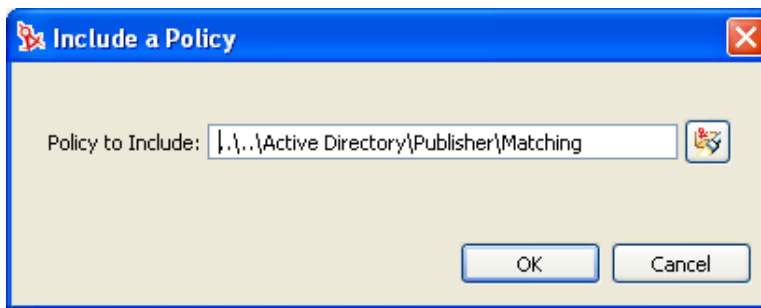
- 2 按一下「瀏覽」圖示。



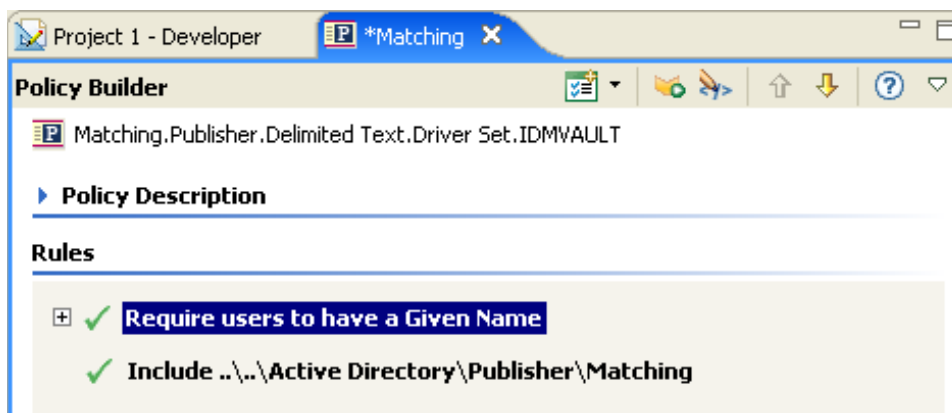
3 瀏覽至要包含的規則，然後按一下「確定」。



4 欄位目前已填入規則的路徑。按一下「確定」。



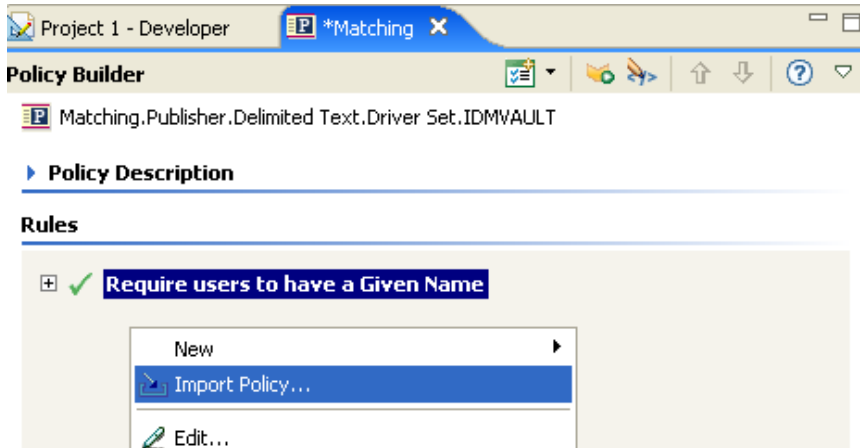
此規則與原始規則的連結。您不能在此位置編輯規則。請存取原始的規則再進行變更。



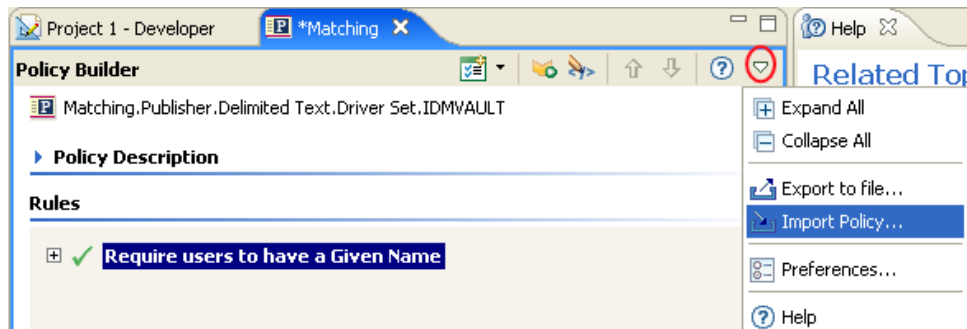
從 XML 檔案輸入規則

規則 (Rule) 和規則 (Policy) 可以儲存為 XML 檔案。若您有包含想要使用之規則 (Rule) 或規則 (Policy) 的檔案，「規則產生器」可讓您輸入該檔案。

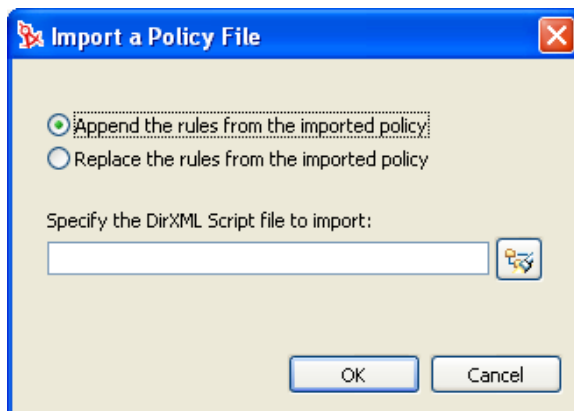
- 1 在「規則產生器」中按一下滑鼠右鍵，並選取「輸入規則」。



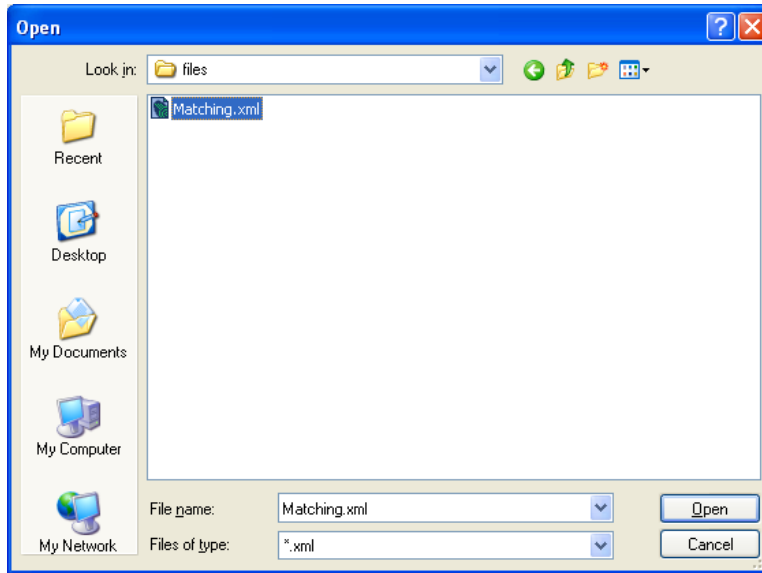
您也可以從工具列的下拉式清單選取「輸入規則」圖示。



- 2 選取下列兩個選項之一：「附加輸入之規則 (Policy) 的規則 (Rule)」或「取代輸入規則 (Policy) 中的規則 (Rule)」。



3 按一下瀏覽圖示並選取包含「DirXML 程序檔」的檔案，然後按一下「開啓」。



4 按一下「確定」。

2.2.4 建立引數

「引數產生器」提供動態的圖形化介面，可讓您建構用於「規則產生器」內的複雜引數運算式。若要存取「引數產生器」，請參閱「[引數產生器](#)」，第 58 頁。

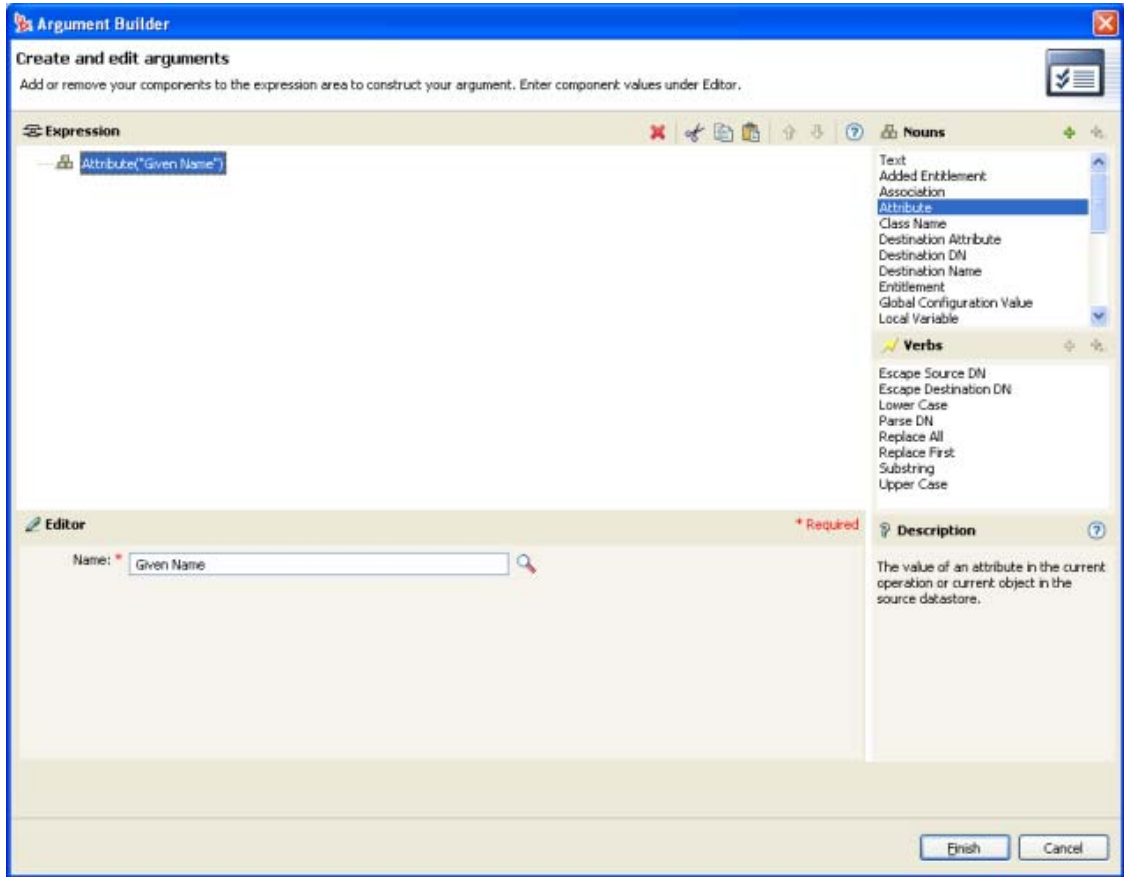
引數由動作動態使用，並且可以衍生自執行時期所展開的記號。

記號可分開成爲兩個類別：名詞和動詞。名詞記號會展開爲衍生自目前操作、來源或目的資料儲存，或是某外部來源的值。動詞記號則會修改從屬於該記號之其他記號的串連結果。

若要定義運算式，請選取一或多個名詞記號（值、物件、變數等），並結合動詞記號（子字串、逸出、大寫和小寫），以建構引數。結合多個記號可建構複雜的引數。

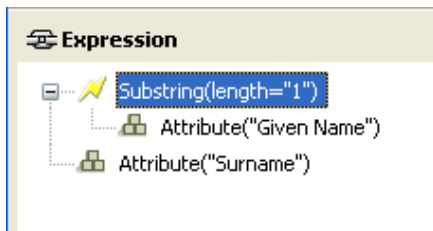
例如，若您要將引數設定為某個屬性值，可以選取屬性名詞，然後選取屬性名稱：

特性 2-2 引數產生器



若您僅想要使用屬性的一部份，可以結合屬性名詞與子字串動詞：

特性 2-3 運算式



在新增名詞或動詞後，您可以在編輯器中提供值，然後立即新增其他的名詞或動詞。您不需要重新整理「運算式」窗格以套用變更；它們會在執行下一個操作時出現。


如需「引數產生器」內可用記號的詳細參考資訊，請參閱「名詞記號」，第 173 頁和「動詞記號」，第 186 頁。

儘管您會使用「引數產生器」定義大多數的引數，但還有幾個產生器會在「規則產生器」內的「條件編輯器」和「動作編輯器」中使用。每個產生器可以遞迴方式呼叫下列清單內的任一個產生器：

- ◆ 「動作產生器」，第 57 頁
- ◆ 「引數產生器」，第 58 頁
- ◆ 「相符屬性產生器」，第 59 頁
- ◆ 「動作引數元件產生器」，第 60 頁
- ◆ 「引數值清單產生器」，第 61 頁
- ◆ 「具名字串產生器」，第 62 頁
- ◆ 「條件引數元件產生器」，第 63 頁
- ◆ 「模式字串產生器」，第 64 頁

下列資訊描述如何存取每個「產生器」。

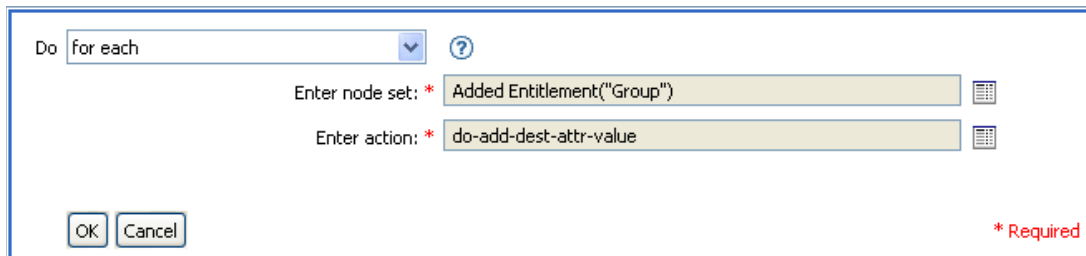
動作產生器


若要啟動「動作產生器」，請選取下列兩項動作之一，然後按一下「編輯引數」圖示 。


- ◆ 針對每個
- ◆ 實作授權


在下列範例中，新增目的屬性值的動作會由將被新增至目前操作的每個「群組」授權所執行。

特性 **2-4** 對於每個動作



Do 

Enter node set: * 

Enter action: * 

* Required

若要定義新增目的屬性值的動作，請按一下啟動「動作產生器」的圖示。在「動作產生器」中，您可以定義所需的動作。在下列範例中，將為每個新增的「群組」授權新增成員屬性至目的物件。

特性 2-5 引數動作產生器

Do: add destination attribute value

Enter attribute name: * Member

Enter class name: Group

Select mode: add to current operation

Select object: DN

Enter DN: * Local Variable("current-node")


Enter value type: String

Enter string: * Destination DN()

OK Cancel

* Required

引數產生器

若要啟動「引數產生器」，請選取下列動作之一，然後按一下「編輯引數」圖示 

- ◆ 「新增關聯」，第 131 頁
- ◆ 「新增目的屬性值」，第 132 頁
- ◆ 「新增目的物件」，第 133 頁
- ◆ 「新增來源屬性值」，第 135 頁
- ◆ 「附加 XML 文字」，第 137 頁
- ◆ 「清除目的屬性值」，第 138 頁 (當選取物件為 DN 或「關聯」時)
- ◆ 「清除來源屬性值」，第 139 頁 (當選取物件為 DN 或「關聯」時)
- ◆ 「刪除目的物件」，第 142 頁 (當選取物件為 DN 或「關聯」時)
- ◆ 「刪除來源物件」，第 143 頁 (當選取物件為 DN 或「關聯」時)
- ◆ 「尋找相符物件」，第 143 頁
- ◆ 「針對每個」，第 144 頁
- ◆ 「移動目的物件」，第 148 頁
- ◆ 「移動來源物件」，第 149 頁
- ◆ 「重新格式化操作屬性」，第 150 頁
- ◆ 「移除關聯」，第 151 頁
- ◆ 「移除目的屬性值」，第 152 頁
- ◆ 「移除來源屬性值」，第 152 頁
- ◆ 「重新命名目的物件」，第 153 頁 (當選取物件為 DN 或「關聯」和「輸入字串」時)
- ◆ 「重新命名來源物件」，第 154 頁 (當選取物件為 DN 或「關聯」和「輸入字串」時)
- ◆ 「設定目的屬性值」，第 158 頁 (當選取物件為 DN 或「關聯」且「輸入值類型」未被結構化時)
- ◆ 「設定目的密碼」，第 159 頁
- ◆ 「設定本地變數」，第 160 頁

- ◆ 「設定操作關聯」，第 161 頁
- ◆ 「設定操作類別名稱」，第 162 頁
- ◆ 「設定操作目的 DN」，第 162 頁
- ◆ 「設定操作內容」，第 163 頁
- ◆ 「設定操作來源 DN」，第 164 頁
- ◆ 「設定操作範本 DN」，第 164 頁
- ◆ 「設定來源屬性值」，第 165 頁
- ◆ 「設定來源密碼」，第 166 頁
- ◆ 「設定 XML 屬性」，第 168 頁
- ◆ 「狀態」，第 168 頁
- ◆ 「追蹤訊息」，第 170 頁

1 使用名詞和動詞建立引數。


您可以結合名詞和動詞，以建立所需的引數。


2 按一下「完成」。

相符屬性產生器


「相符屬性產生器」可讓您選取「尋找相符物件」，第 143 頁動作所使用的屬性及值，以判定資料儲存內是否有相符的物件。


例如，若您想要根據公用名稱和位置來比對使用者：

- 1 選取「尋找相符物件」動作。
- 2 選取搜尋相符物件的範圍。從「項目」、「從屬物件」或「子網路樹」中選取。
- 3 指定搜尋起始點的 DN。
- 4 按一下「編輯相符屬性」圖示，以啟動「相符屬性產生器」。

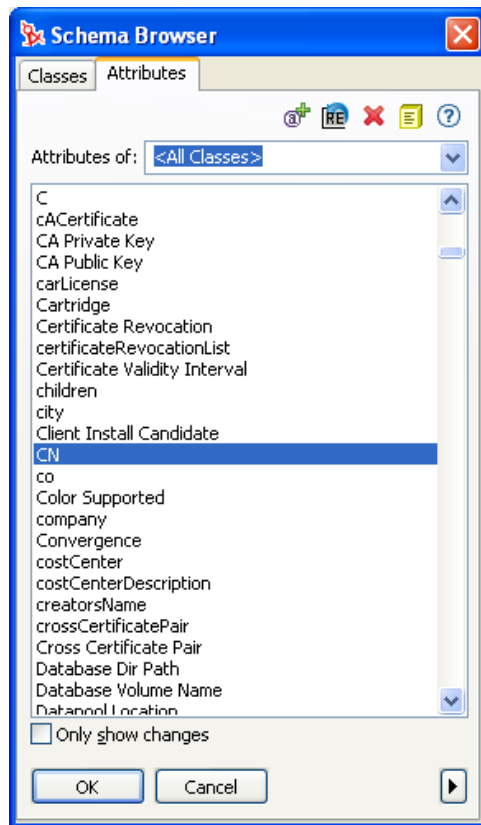
Do 

Select scope:

Enter DN: 


Enter match attributes: 

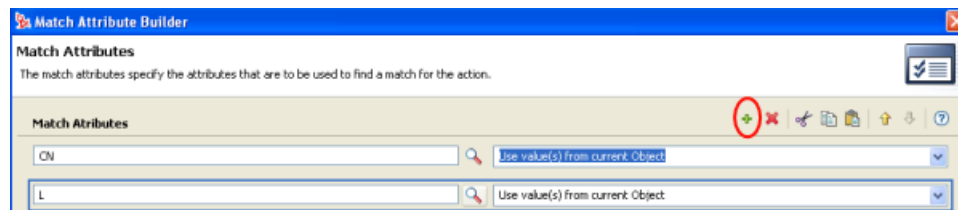
5 Click the *Browse attributes*  icon to launch the Schema Browser.



6 按一下「屬性」索引標籤，然後瀏覽至所需的屬性並加以選取。

7 按一下「確定」。

若您想要新增一個以上的屬性，請按一下「附加新項目」圖示 ，以新增另一行。



8 按一下「完成」。


動作引數元件產生器


若要啟動「動作引數元件產生器」，請在「輸入值類型」選項為「結構化」時，選取下列動作之一，然後按一下「編輯元件」圖示 .


- ◆ 「新增目的屬性值」，第 132 頁
- ◆ 「新增來源屬性值」，第 135 頁
- ◆ 「重新格式化操作屬性」，第 150 頁
- ◆ 「移除目的屬性值」，第 152 頁


- ◆ 「移除來源屬性值」，第 152 頁
- ◆ 「設定目的屬性值」，第 158 頁
- ◆ 「設定來源屬性值」，第 165 頁


特性 2-6 新增目的屬性值動作


Do 


Enter attribute name: * 


Enter class name: 


Select mode: 

Select object: 

Enter DN: * 

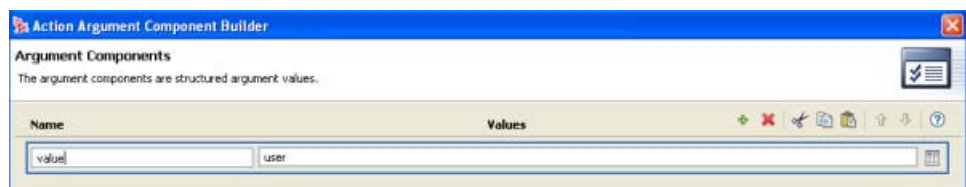
Enter value type: 

Enter components: * 

1 當值類型設定為結構化時，按一下「編輯元件」圖示 。


2 建立動作元件的值。

您可以輸入值或按一下「編輯引數」 圖示，以在「引數產生器」內建立值。




3 按一下「完成」。


引數值清單產生器


若要啟動「引數值清單產生器」，請選取下列動作，然後按一下「編輯引數」圖示 。


- ◆ 設定預設的屬性值

特性 2-7 設定預設的屬性值


Do 

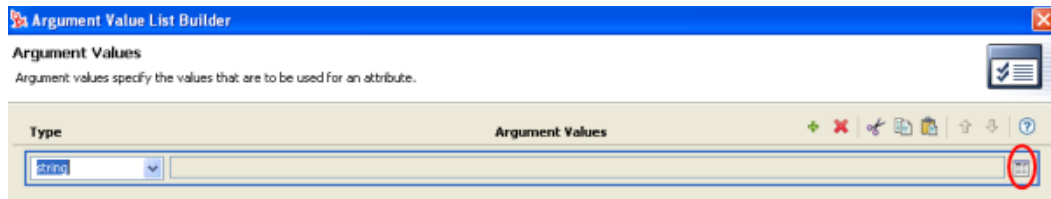
Enter attribute name: * 


Write back: 

Enter argument values: * 

1 選取值的類型：*counter*、*dn*、*int*、*interval*、*octet*、*state*、*string*、*structured*、*teleNumber*、*time*。

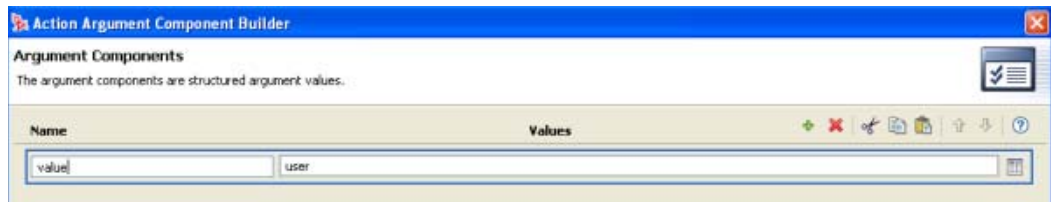
- 2 按一下「編輯值清單」圖示 。



- 3 按一下「編輯引數」圖示 。


- 4 建立動作元件的值。

您可以輸入值或按一下「編輯引數」圖示，以在「引數產生器」內建立值。




- 5 按一下「完成」。

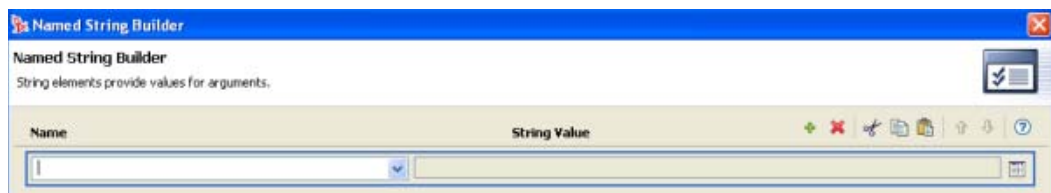
具名字串產生器

若要啟動「具名字串產生器」，請選取下列動作之一，然後按一下「編輯字串」圖示 。

- ◆ 產生事件
- ◆ 傳送電子郵件
- ◆ 從範本傳送電子郵件

- 1 從下拉式清單中選取字串的名稱。

- 2 按一下「編輯引數」圖示 ，以啟動「引數產生器」來建立字串值。



- 3 按一下「完成」。


對於「傳送電子郵件」動作而言，具名字串會對應至電子郵件的元素：

特性 **2-8** 傳送電子郵件動作內的電子郵件元素





對應至啟動「具名字串產生器」動作的說明檔案包含可能的值之完整清單。

條件引數元件產生器


若要啟動「條件引數元件產生器」，請選取下列條件之一，然後您必須選取「模式」的結構化選項，才能看到「啟動 *ArgComponent* 產生器」圖示 。

- ◆ If Attribute
- ◆ If Destination Attribute
- ◆ If Source Attribute
- ◆ If Operation Attribute

Condition 

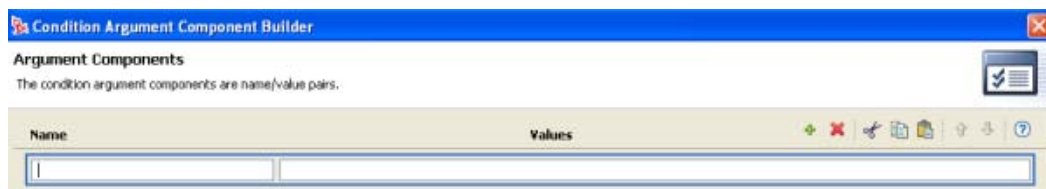
Name * 

Operator *

Mode 

Value

1 指定條件元件的名稱和值。

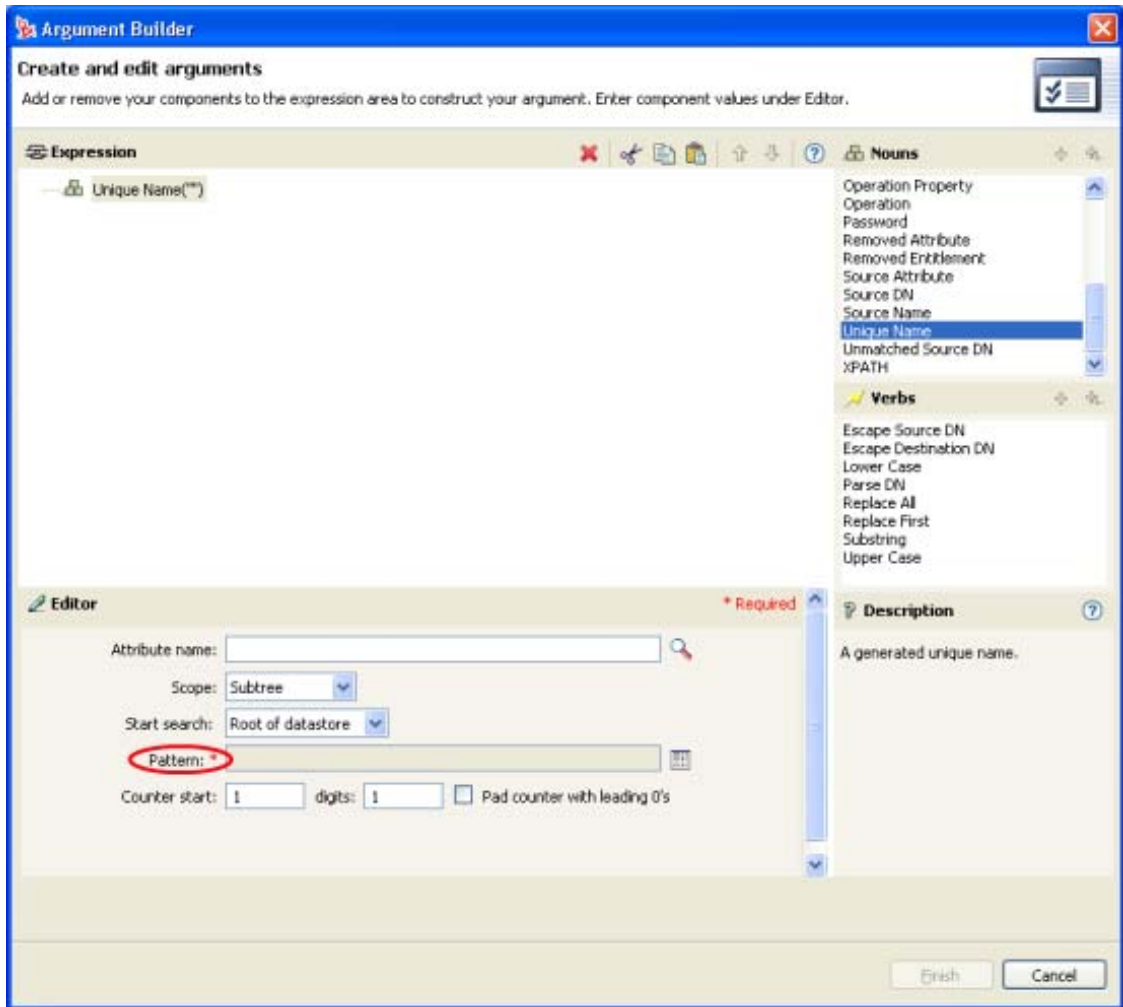




2 按一下「完成」。

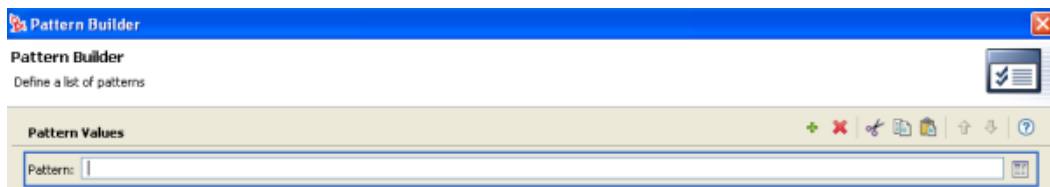
模式字串產生器

您可以在選取「**唯一名稱**」記號時，從「引數產生器」編輯器啟動「模式字串產生器」。「引數產生器」編輯器窗格會顯示「模式」欄位，您可以按一下該欄位啟動「模式字串產生器」。

特性 2-9 引數產生器內的唯一名稱記號



- 1 按一下「編輯模式」圖示 ，以啟動「模式產生器」。
- 2 指定模式或按一下「編輯引數」圖示 ，以使用「引數產生器」建立模式。

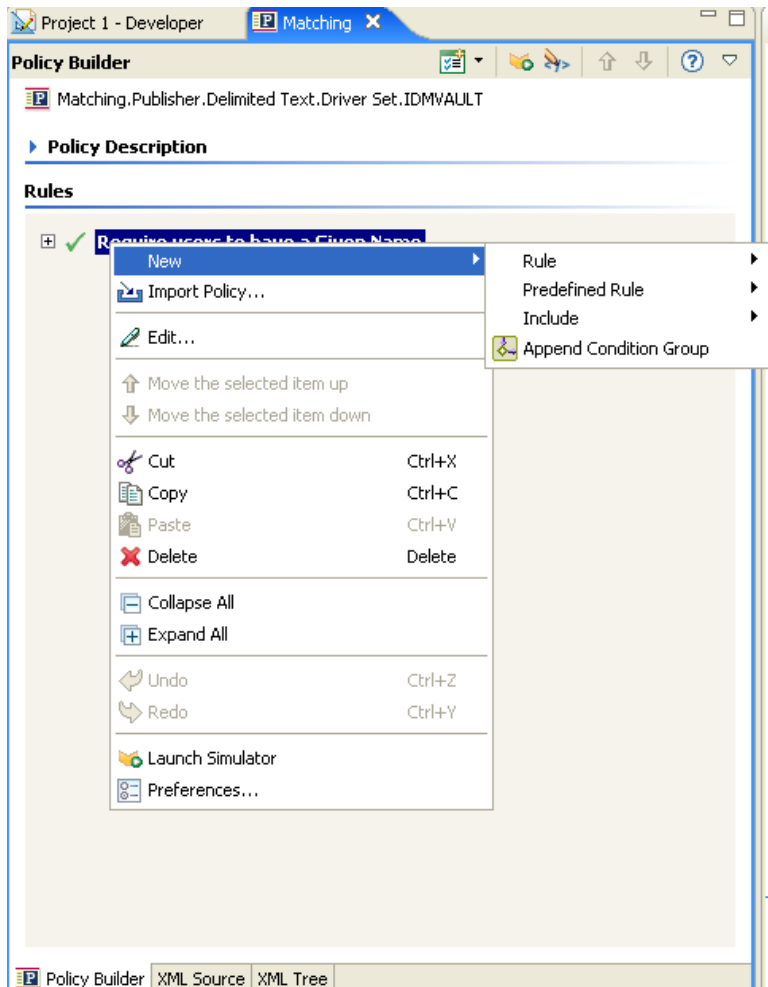


- 3 按一下「完成」。

2.2.5 編輯規則

「規則產生器」可讓您建立及編輯規則。您可以拖放規則 (Rule)、條件和動作。如需其他的操作，請使用「規則產生器」工具列。若要顯示快顯功能表，請以滑鼠右鍵按一下某項目。

特性 2-10 規則產生器的快顯功能表及工具列



規則產生器內的動作和功能表項目

下表包含「規則產生器」內不同動作和功能表項目的清單。

表格 2-3 規則產生器的動作和功能表項目

操作	描述
全部折疊	將所有展開的規則折疊起來。
複製	將選取的项目複製到「剪貼簿」。
複製並放下	選取项目、按 Ctrl 鍵，然後拖曳项目。
剪下	將選取的项目剪下並複製到「剪貼簿」。

操作	描述
刪除	刪除選取的項目。
停用	停用規則、條件或動作。按一下  圖示。
拖放	讓您選取某項目，然後為其重新定位。選取項目，然後將其拖曳至新位置。
編輯	可讓您編輯選取的項目。若要開啓「規則產生器」，請選取規則，然後按一下「編輯」。
啓用	啓用規則、條件或動作。按一下  圖示。
全部展開	展開所有規則，使您能夠檢視每個規則的條件和動作。
輸入規則	從檔案系統輸入規則，並將其附加至規則中，或取代規則 (Policy) 的所有規則 (Rule)。
啓動模擬器	啓動「規則模擬器」。
移動並放下	讓您選取及移動項目。選取項目，然後拖曳該項目。
將選取的項目向下移動	將項目於規則清單中下移。
將選取的項目向上移動	將項目於規則清單中上移。
新增 > 條件群組	在選取項目之後新增條件群組。
新增 > 包含項目	在選取項目之後建立新的包含項目。
新增 > 預先定義的規則	插入預先定義的規則。
新增 > 規則	在選取項目之後建立新規則。
貼上	將「剪貼簿」的內容貼在選取項目之後。
優先設定	可讓您變更資訊的顯示方式。
選取	按一下任意項目來加以選取。

鍵盤支援

您可以使用按鍵和滑鼠瀏覽「規則產生器」。支援的按鍵列於下方。

表格 2-4 規則產生器的鍵盤支援

按鍵	描述
Ctrl+C	將選取的項目複製到「剪貼簿」。
Ctrl+X	將選取的項目剪下並新增到「剪貼簿」。
Ctrl+V	將「剪貼簿」的內容貼至選取項目之後。
刪除	刪除選取的項目。
向左鍵	折疊某個規則節點。
向右鍵	展開某個規則節點。

按鍵	描述
向上鍵	向上導覽。
向下鍵	向下導覽。
Ctrl+Z	復原
Ctrl+Y	取消復原

重新命名規則

- 1 在「大綱」檢視窗中，選取要重新命名的規則。
- 2 按一下滑鼠右鍵，並選取「內容」。
- 3 在「規則名稱」欄位內變更規則的名稱。

- 4 按一下「確定」。

儲存您的工作

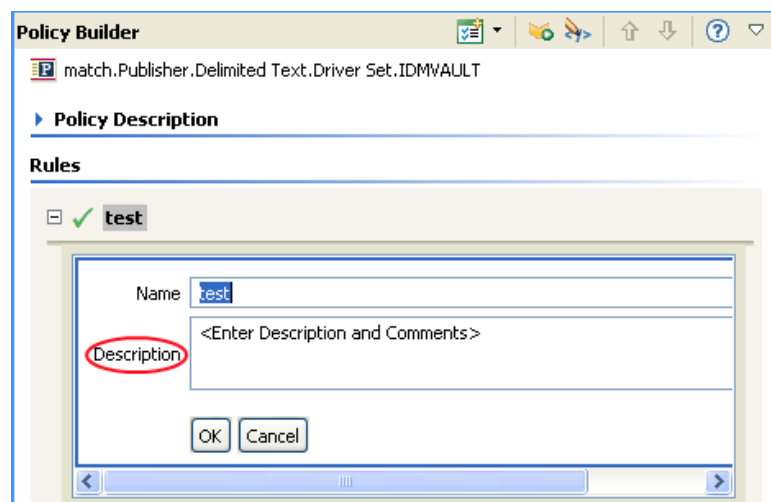
請進行下列其中一項操作：

- ◆ 從「主功能表」中，按一下「檔案」>「儲存」（或「全部儲存」）。
- ◆ 在編輯器索引標籤內按一下 *X*，關閉編輯器。
- ◆ 從「主功能表」的檔案功能表中選取「關閉」。
- ◆ 按 Ctrl+S。

規則描述

「描述」欄位提供可以新增與規則功能相關之註解的地方。

特性 **2-11** 規則描述



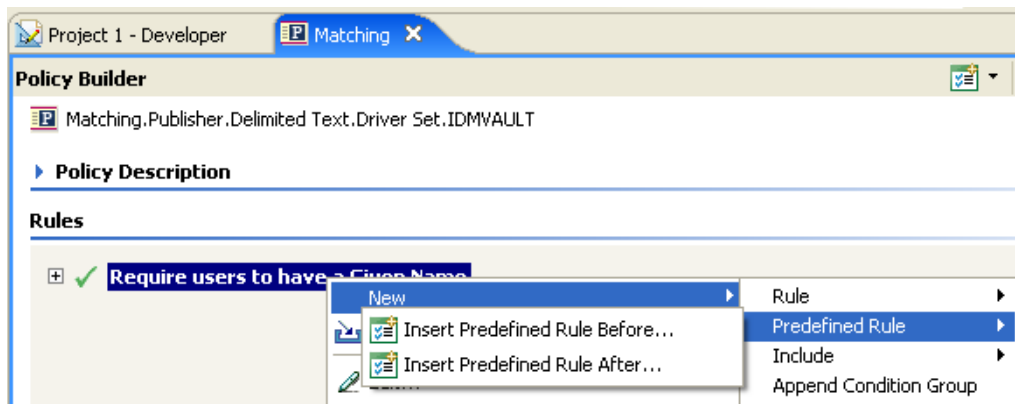
2.2.6 使用預先定義的規則

Designer 包含二十個預先定義的規則。您可以輸入及使用這些規則，並建立自己的規則。這些規則包含管理員常用的任務。您必須提供環境特定的資訊，以藉此自定規則。

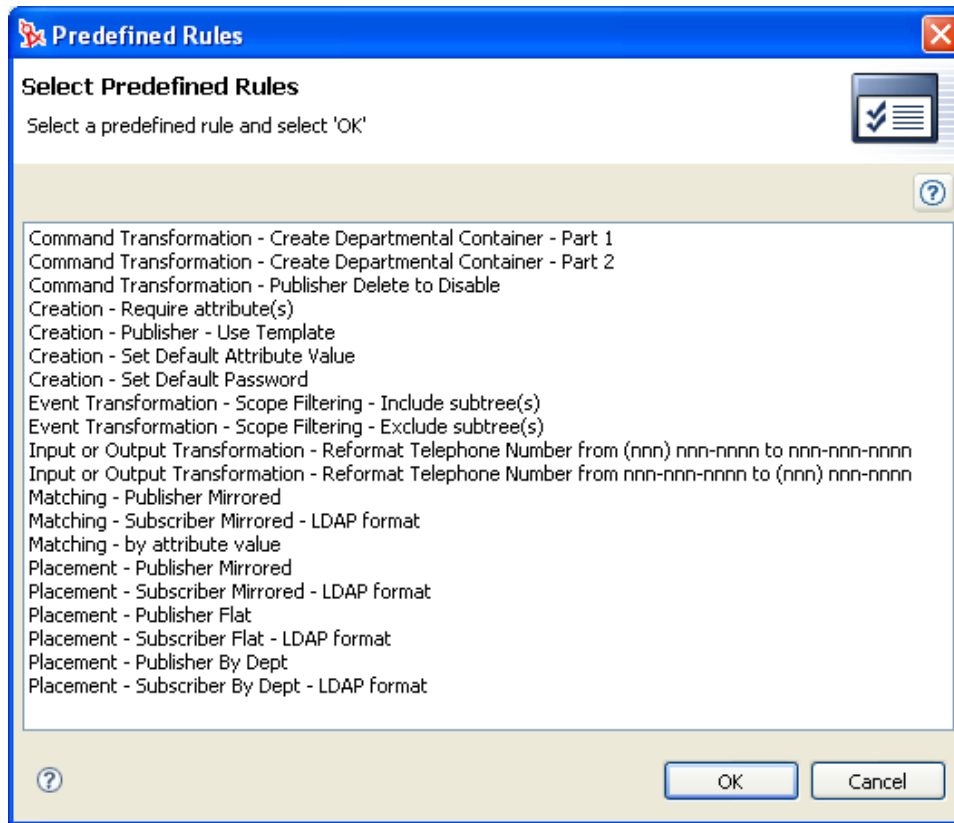
- ◆ 「指令轉換：建立部門容器：第 1 部份和第 2 部份」，第 69 頁
- ◆ 「指令轉換：要停用的發行者刪除」，第 71 頁
- ◆ 「建立：需要屬性」，第 73 頁
- ◆ 「建立：發行者：使用範本」，第 74 頁
- ◆ 「建立：設定預設屬性值」，第 75 頁
- ◆ 「建立：設定預設密碼」，第 77 頁
- ◆ 「事件轉換：範圍過濾：包括子網路樹」，第 78 頁
- ◆ 「事件轉換：範圍過濾：排除子網路樹」，第 80 頁
- ◆ 「輸入或輸出轉換：將電話號碼從 (nnn) nnn-nnnn 重新格式化為 nnn-nnn-nnnn」，第 81 頁
- ◆ 「輸入或輸出轉換：將電話號碼從 nnn-nnn-nnnn 重新格式化為 (nnn) nnn-nnnn」，第 82 頁
- ◆ 「比對：鏡像複製發行者」，第 83 頁
- ◆ 「比對：鏡像複製訂閱者：LDAP 格式」，第 85 頁
- ◆ 「比對：依屬性值」，第 87 頁
- ◆ 「佈置：鏡像複製發行者」，第 88 頁
- ◆ 「佈置：鏡像複製訂閱者：LDAP 格式」，第 90 頁
- ◆ 「佈置：一般發行者」，第 91 頁
- ◆ 「佈置：一般訂閱者：LDAP 格式」，第 93 頁
- ◆ 「佈置：依部門的發行者」，第 94 頁
- ◆ 「佈置：依部門的訂閱者：LDAP 格式」，第 96 頁

若要使用預先定義的規則：

- 1 在「規則產生器」中按一下滑鼠右鍵並選取「新增」>「預先定義的規則」>「在之前插入預先定義的規則」或「在之後插入預先定義的規則」。



「預先定義的規則」對話方塊會顯示可用的規則清單。




指令轉換：建立部門容器：第 1 部份和第 2 部份

在目的資料儲存中建立部門容器（若其中不存在該容器）。在驅動程式內的「指令轉換」規則 (Policy) 上實作規則 (Rule)。您可以在「訂閱者」或「發行者」其中一個通道上實作規則，或同時在兩個通道上實作規則。

使用預先定義的規則包含兩個步驟：在「指令轉換」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「指令轉換」規則 (Policy)，請跳至「輸入預先定義的規則」，第 70 頁。

建立規則

- 1 從「大綱」檢視窗或「規則流程」檢視窗中，選取「發行者」或「訂閱者」通道。
- 2 在「規則集」檢視窗中選取「指令轉換」規則集，然後按一下「建立或新增新規則至規則集」圖示 ，以建立新規則。
- 3 按一下「建立新規則」，然後再按「下一步」。
- 4 為規則命名。

5 使用填入的位置來將規則放入驅動程式內。

6 選取「建立規則後開啓編輯器」，然後再按「下一步」。

7 選取「DirXML 指令檔」為規則類型，然後按一下「完成」。

8 此時會出現檔案衝突視窗，其中內含「在編輯此項目前必須先儲存。是否要儲存編輯器的變更並繼續？」訊息。請按一下「是」。此時會啟動「規則產生器」，並且會儲存新的「指令轉換」規則。

輸入預先定義的規則

1 在「規則產生器」內按一下滑鼠右鍵，然後按一下「新增」>「預先定義的規則」>「在之前插入預先定義的規則」或「在之後插入預先定義的規則」。

2 選取「指令轉換：建立部門容器：第 1 部份」，然後按一下「確定」。

3 在「規則產生器」內按一下滑鼠右鍵，然後按一下「新增」>「預先定義的規則」>「在之前插入預先定義的規則」或「在之後插入預先定義的規則」。

4 選取「指令轉換：建立部門容器：第 2 部份」，然後按一下「確定」。

5 按一下「檔案」>「儲存」以儲存規則。

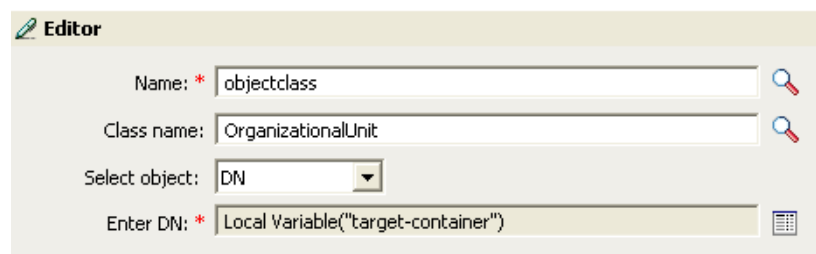
您環境特有的規則中沒有要變更的資訊。

重要：請確定規則依順序列出。第 1 部份必須在第 2 部份之前執行。

規則的運作方式

當某個物件的目的位置不存在時，可以使用此規則。此規則會先建立容器，然後將物件放在容器中，不會因為不能放置物件而遭到否決。

第 1 部份會尋找任何「新增」事件。當發生「新增」事件時，會設定兩個本地變數。第一個本地變數會被命名為 `target-container`。`target-container` 的值會被設為目的 DN。第二個本地變數會被命名為 `does-target-exist`。`does-target-exist` 的值會被設為 `objectclass` 的目的屬性值。類別會被設定為 `OrganizationalUnit`。`OrganizationalUnit` 的 DN 會被設定為 `target-container` 的本地變數。



The screenshot shows a configuration window titled "Editor". It contains four input fields:

- Name: * objectclass
- Class name: OrganizationalUnit
- Select object: DN (dropdown menu)
- Enter DN: * Local Variable("target-container")


第 2 部份會檢查是否可以使用本地變數 `does-target-exist`。第 2 部份同時也會檢查本地變數 `does-target-exist` 的值是否設定為空白值。若為空白值，則會建立一個「組織單位」物件。組織單位的 DN 會被設定為本地變數 `target-container` 的值。同時也會新增 OU 屬性的值。OU 屬性的值會被設定為 `target-container` 的本地變數。它會使用來源格式為目的 DN，而目的格式採用點格式。

指令轉換：要停用的發行者刪除

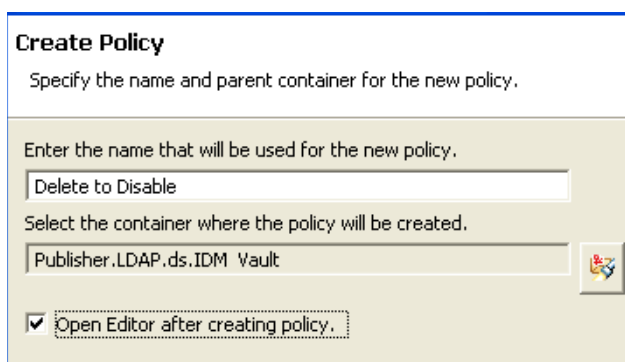
將使用者的「刪除」事件轉換為停用使用者物件。在驅動程式內的「指令轉換」規則 (Policy) 上實作規則 (Rule)。您必須在「發行者」通道上實作規則。

使用預先定義的規則包含兩個步驟：在「指令轉換」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「指令轉換」規則 (Policy)，請跳至「輸入預先定義的規則」，第 72 頁。

建立規則

- 1 從「大綱」檢視窗或「規則流程」檢視窗中，選取「發行者」通道。
- 2 在「規則集」檢視窗中選取「指令轉換」規則集，然後按一下「建立或新增新規則至規則集」圖示 ，以建立新規則。
- 3 按一下「建立新規則」，然後再按「下一步」。
- 4 為規則命名。

5 使用填入的位置來將規則放入驅動程式內。



6 選取「建立規則後開啓編輯器」，然後再按「下一步」。

7 選取「DirXML 指令檔」為規則類型，然後按一下「完成」。

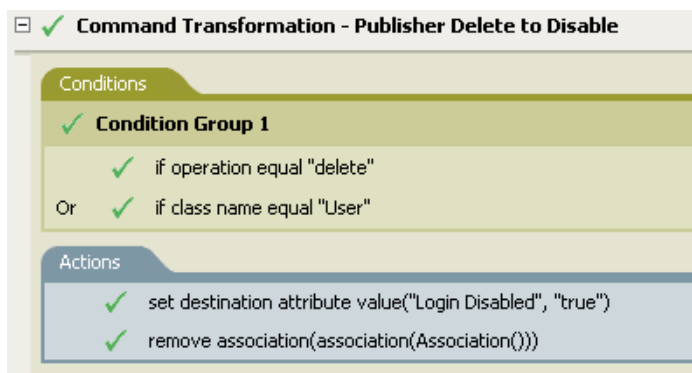
8 此時會出現檔案衝突視窗，其中內含「在編輯此項目前必須先儲存。是否要儲存編輯器的變更並繼續？」訊息。請按一下「是」。此時會啟動「規則產生器」，並且會儲存新的「指令轉換」規則。

輸入預先定義的規則

1 在「規則產生器」內按一下滑鼠右鍵，然後按一下「新增」>「預先定義的規則」>「在之前插入預先定義的規則」或「在之後插入預先定義的規則」。

2 選取「指令轉換：要停用的發行者刪除」，然後按一下「確定」。

3 按一下「檔案」>「儲存」以儲存規則。



您環境特有的規則中沒有要變更的資訊。

規則的運作方式


當已連接資料儲存內發生「刪除」事件時，可以使用此規則。以停用「使用者」物件的方式取代刪除 Identity Vault 內的使用者物件。每次發生「使用者」物件的「刪除」事件時，Login Disabled 的目的屬性值會被設定為 True，並且會移除與「使用者」物件的關聯。「使用者」物件無法再登入至 Novell eDirectory 網路樹，但「使用者」物件未被刪除。

建立：需要屬性

此規則不允許在未填入必要屬性的情形下建立使用者物件。在驅動程式內的「建立」規則 (Policy) 上實作規則 (Rule)。您可以在「訂閱者」或「發行者」其中一個通道上實作規則，或同時在兩個通道上實作規則。

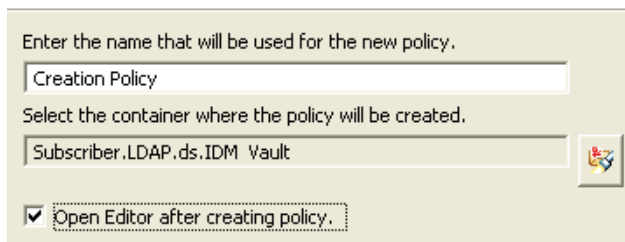
使用預先定義的規則包含兩個步驟：在「建立」規則集中建立規則 (Policy)，以及輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「建立」規則 (Policy)，請跳至「輸入預先定義的規則」，第 73 頁。

建立規則

- 1 從「大綱」檢視窗或「規則流程」檢視窗中，選取「發行者」或「訂閱者」通道。
- 2 在「規則集」檢視窗中選取「建立」規則集，然後按一下「建立或新增新規則至規則集」圖示 ，以建立新規則。
- 3 按一下「建立新規則」，然後再按「下一步」。
- 4 為規則命名。
- 5 使用填入的位置來將規則放入驅動程式內。

Create Policy

Specify the name and parent container for the new policy.



Enter the name that will be used for the new policy.

Creation Policy

Select the container where the policy will be created.

Subscriber.LDAP.ds.IDM Vault

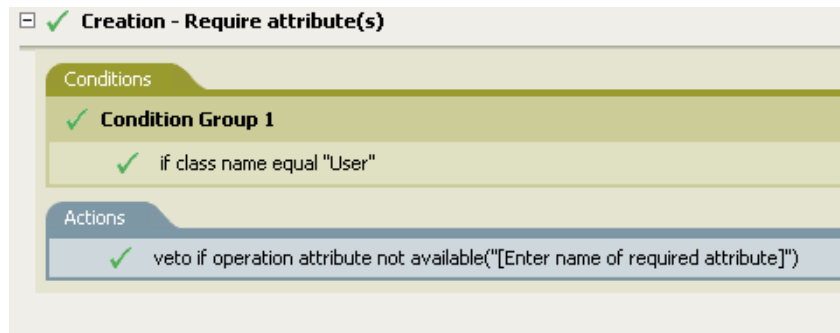
Open Editor after creating policy.

- 6 建立規則後選取「開啓編輯器」，然後再按「下一步」。
- 7 選取「DirXML 指令檔」為規則類型，然後按一下「完成」。
- 8 此時會出現檔案衝突視窗，其中內含「" 在編輯此項目前必須先儲存。是否要儲存編輯器的變更並繼續？"」訊息。請按一下「是」。此時會啟動「規則產生器」，並且會儲存新的「建立」規則。

輸入預先定義的規則

- 1 在「規則產生器」內按一下滑鼠右鍵，然後按一下「新增」>「預先定義的規則」>「在之前插入預先定義的規則」或「在之後插入預先定義的規則」。
- 2 選取「建立：需要屬性」，然後按一下「確定」。
- 3 連按兩下「動作」索引標籤，以編輯動作。
- 4 從「輸入名稱欄位」刪除「[輸入必要屬性的名稱]」。
- 5 瀏覽至要建立之「使用者」物件所需的屬性，然後按一下「確定」。
- 6 按一下「確定」。

7 選取「檔案」>「儲存」以儲存規則。



規則的運作方式

如果您的企業流程需要使用者在建立使用者物件時填入特定屬性，就使用此規則。在建立了使用者物件時，除非已提供必要的屬性，否則此規則會否決物件的建立。您可以擁有一或多個必要的屬性。


若您需要一個以上的必要屬性，請在動作上按一下滑鼠右鍵，然後選取「新增 > 附加動作」。選取「如果操作屬性無法使用則否決」，然後瀏覽至所需的屬性。

建立：發行者：使用範本

允許在建立「使用者」物件期間使用 Novell eDirectory 範本物件。在驅動程式內的「發行者建立」規則 (Policy) 上實作規則 (Rule)。您只可以在「發行者」通道上實作規則。

使用預先定義的規則包含兩個步驟：在「建立」規則集中建立規則 (Policy)，以及輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「建立」規則 (Policy)，請跳至「輸入預先定義的規則」，第 75 頁。

建立規則

- 1 從「大綱」檢視窗或「規則流程」檢視窗中，選取「發行者」通道。
- 2 在「規則集」檢視窗中選取「建立」規則集，然後按一下「建立或新增新規則至規則集」圖示 ，以建立新規則。
- 3 按一下「建立新規則」，然後再按「下一步」。
- 4 為規則命名。
- 5 使用填入的位置來將規則放入驅動程式內。


Create Policy

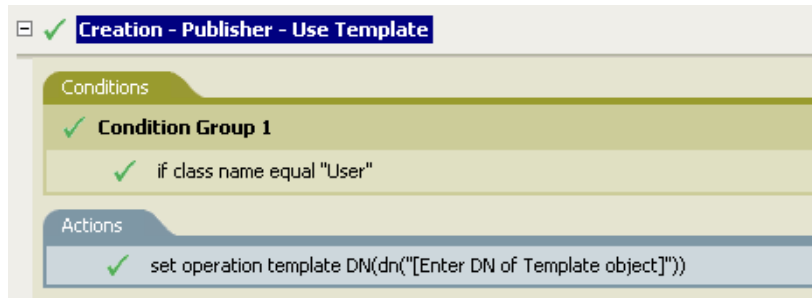
Specify the name and parent container for the new policy.

- 6 選取「建立規則後開啓編輯器」，然後再按「下一步」。

- 7 選取「DirXML 指令檔」為規則類型，然後按一下「完成」。
- 8 此時會出現檔案衝突視窗，其中內含「在編輯此項目前必須先儲存。是否要儲存編輯器的變更並繼續？」訊息。請按一下「是」。此時會啟動「規則產生器」，並且會儲存新的「建立」規則。

輸入預先定義的規則

- 1 在「規則產生器」內按一下滑鼠右鍵，然後按一下「新增」>「預先定義的規則」>「在之前插入預先定義的規則」或「在之後插入預先定義的規則」。
- 2 選取「建立：發行者：使用範本」，然後按一下「確定」。
- 3 連按兩下「動作」索引標籤，以編輯動作。
- 4 從「輸入 DN」欄位中刪除「[輸入範本物件的 DN]」。
- 5 按一下「編輯引數圖示」，以啟動「引數產生器」。
- 6 在「名詞」清單中選取「文字」。
- 7 連按兩下「文字」將其新增至引數中。
- 8 在「編輯器」中按一下瀏覽圖示，瀏覽至範本物件並加以選取，然後按一下「確定」。
- 9 按一下「確定」。
- 10 按一下「檔案」>「儲存」以儲存規則。



規則的運作方式

當您要在 Identity Vault 內使用範本物件來建立使用者時，可以使用此規則。若您的不同使用者擁有相同的屬性，可使用範本來節省時間。您在範本物件內填入資訊，然後在建立「使用者」物件時，Identity Manager 會呼叫範本，並使用該範本來建立「使用者」物件。

在建立「使用者」物件期間，此規則會執行設定操作範本 DN 的動作。此動作會呼叫範本物件，並以範本內的資訊建立「使用者」物件。


建立：設定預設屬性值

可讓您設定建立「使用者」物件期間指定之屬性的預設值。您可以在驅動程式內的「訂閱者建立」規則 (Policy) 或「發行者建立」規則 (Policy) 上實作規則 (Rule)。

使用預先定義的規則包含兩個步驟：在「建立」規則集中建立規則 (Policy)，以及輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「建立」規則 (Policy)，請跳至「輸入預先定義的規則」，第 76 頁。

建立規則


- 1 從「大綱」檢視窗或「規則流程」檢視窗中，選取「發行者」或「訂閱者」通道。

- 2 在「規則集」檢視窗中選取「建立」規則集，然後按一下「建立或新增新規則至規則集」圖示 ，以建立新規則。
- 3 按一下「建立新規則」，然後再按「下一步」。
- 4 為規則命名。
- 5 使用填入的位置來將規則放入驅動程式內。

Create Policy



Specify the name and parent container for the new policy.

Enter the name that will be used for the new policy.

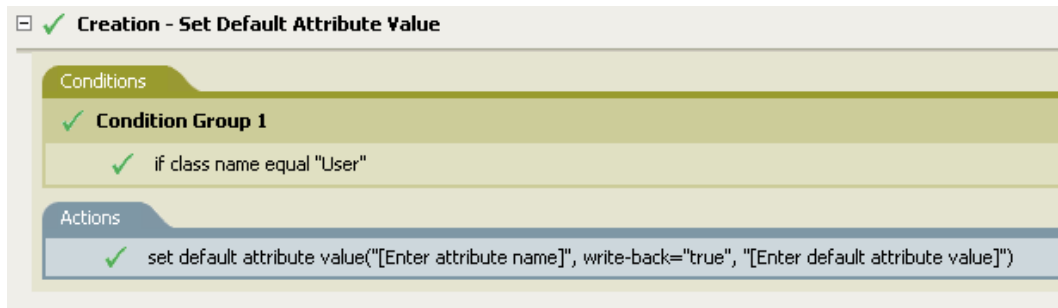
 Select the container where the policy will be created.
 
 Open Editor after creating policy.

- 6 選取「建立規則後開啓編輯器」，然後再按「下一步」。
- 7 選取「DirXML 指令檔」為規則類型，然後按一下「完成」。
- 8 此時會出現檔案衝突視窗，其中內含「"在編輯此項目前必須先儲存。是否要儲存編輯器的變更並繼續？」訊息。請按一下「是」。此時會啓動「規則產生器」，並且會儲存新的「建立」規則。

輸入預先定義的規則

- 1 在「規則產生器」內按一下滑鼠右鍵，然後按一下「新增」>「預先定義的規則」>「在之前插入預先定義的規則」或「在之後插入預先定義的規則」。
- 2 選取「建立：設定預設屬性值」，然後按一下「確定」。
- 3 連按兩下「動作」索引標籤，以編輯動作。
- 4 從「輸入屬性名稱欄位」中刪除「[輸入屬性名稱]」。
- 5 按一下瀏覽圖示，然後瀏覽至想要建立的屬性並加以選取。
- 6 從「輸入引數值」欄位中刪除「[輸入預設屬性值]」。
- 7 按一下「編輯引數」圖示 ，以啓動「引數值清單產生器」。
- 8 選取要為值設定的資料類型。
- 9 按一下「編輯引數」圖示 ，以啓動「引數產生器」。
- 10 在「引數產生器」中為屬性建立值，然後按一下「確定」。
- 11 按一下「確定」。

12 按一下「檔案」>「儲存」以儲存規則。



規則的運作方式

當您要以預設的屬性和值建立「使用者」物件時，可以使用此規則。在建立「使用者」物件時，此規則會設定屬性和屬性的值。


若您想要定義一個以上的屬性值，請在動作上按一下滑鼠右鍵，然後按一下「新增」>「附加動作」。選取動作、設定預設的屬性值，並遵循[步驟 1, 第 76 頁](#)到[步驟 12, 第 77 頁](#)來指定屬性值。

建立：設定預設密碼

在建立使用者物件期間，為使用者物件設定預設的密碼。在驅動程式內的「建立」規則 (Policy) 上實作規則 (Rule)。您可以在「訂閱者」或「發行者」其中一個通道上實作規則，或同時在兩個通道上實作規則。

使用預先定義的規則包含兩個步驟：在「建立」規則集中建立規則 (Policy)，以及輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「建立」規則 (Policy)，請跳至[「輸入預先定義的規則」, 第 78 頁](#)。

建立規則

- 1 從「大綱」檢視窗或「規則流程」檢視窗中，選取「發行者」或「訂閱者」通道。
- 2 在「規則集」檢視窗中選取「建立」規則集，然後按一下「建立或新增新規則至規則集」圖示 ，以建立新規則。
- 3 按一下「建立新規則」，然後再按「下一步」。
- 4 為規則命名。
- 5 使用填入的位置來將規則放入驅動程式內。

Create Policy

Specify the name and parent container for the new policy.

Enter the name that will be used for the new policy.
Creation Policy

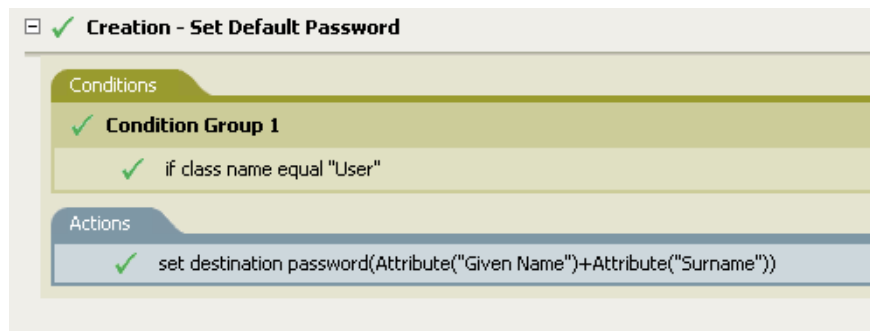
Select the container where the policy will be created.
Subscriber.LDAP.ds.IDM Vault

Open Editor after creating policy.

- 6 選取「建立規則後開啓編輯器」，然後再按「下一步」。
- 7 選取「DirXML 指令檔」為規則類型，然後按一下「完成」。
- 8 此時會出現檔案衝突視窗，其中內含「"在編輯此項目前必須先儲存。是否要儲存編輯器的變更並繼續?"」訊息。請按一下「是」。此時會啓動「規則產生器」，並且會儲存新的「建立」規則。

輸入預先定義的規則

- 1 在「規則產生器」內按一下滑鼠右鍵，然後按一下「新增」>「預先定義的規則」>「在之前插入預先定義的規則」或「在之後插入預先定義的規則」。
- 2 選取「建立：設定預設密碼」，然後按一下「確定」。
- 3 按一下「檔案」>「儲存」以儲存規則。



您環境特有的規則中沒有要變更的資訊。

規則的運作方式

當您要以預設的密碼建立「使用者」物件時，可以使用此規則。在「使用者」物件建立期間，會以「使用者」物件的 Given Name 屬性加上 Surname 屬性為「使用者」物件設定密碼。


您可以透過編輯引數的方式變更預設的密碼值。您可以透過「引數產生器」將密碼設定為任何其他值。

事件轉換：範圍過濾：包括子網路樹

除了指定的子網路樹之外，排除所有發生的事件。在驅動程式內的「事件轉換」規則 (Policy) 上實作規則 (Rule)。您可以在「訂閱者」或「發行者」其中一個通道上實作規則，或同時在兩個通道上實作規則。

使用預先定義的規則包含兩個步驟：在「事件轉換」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「事件轉換」規則 (Policy)，請跳至[輸入預先定義的規則 \(第 79 頁\)](#)。

建立規則

- 1 從「大綱」檢視窗或「規則流程」檢視窗中，選取「發行者」或「訂閱者」通道。
- 2 在「規則集」檢視窗中選取「事件轉換」規則集，然後按一下「建立或新增新規則至規則集」圖示 ，以建立新規則。
- 3 按一下「建立新規則」，然後再按「下一步」。

- 4 為規則命名。
- 5 使用填入的位置來將規則放入驅動程式內。

- 6 選取「建立規則後開啓編輯器」，然後再按「下一步」。
- 7 選取「DirXML 指令檔」為規則類型，然後按一下「完成」。
- 8 此時會出現檔案衝突視窗，其中內含「"在編輯此項目前必須先儲存。是否要儲存編輯器的變更並繼續？」訊息。請按一下「是」。此時會啟動「規則產生器」，並且會儲存新的「事件轉換」規則。

輸入預先定義的規則

- 1 在「規則產生器」內按一下滑鼠右鍵，然後選取「新增」>「預先定義的規則」>「在之前插入預先定義的規則」或「在之後插入預先定義的規則」。
- 2 選取「事件轉換：範圍過濾：包括子網路樹」，然後按一下「確定」。
- 3 連按兩下「條件」索引標籤，以編輯條件。
- 4 刪除「值」欄位內的「[輸入要包含的子網路樹]」。
- 5 按一下瀏覽按鈕，瀏覽想要同步化事件的部份 Identity Vault 網路樹，然後按一下「確定」。
- 6 按一下「確定」。
- 7 按一下「檔案」>「儲存」以儲存規則。

規則的運作方式


如果您希望在同步化時，排除部份 Identity Vault，可以使用此規則。它可讓您不需使用「過濾器」的情形下同步化某些物件而排除其他物件。當某事件未在 Identity Vault 的指定部份發生時，該事件將會被否決。

事件轉換：範圍過濾：排除子網路樹

排除發生在指定子網路樹內的所有事件。在驅動程式內的「事件轉換」規則 (Policy) 上實作規則 (Rule)。您可以在「訂閱者」或「發行者」其中一個通道上實作規則，或同時在兩個通道上實作規則。

使用預先定義的規則包含兩個步驟：在「事件轉換」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「事件轉換」規則 (Policy)，請跳至「輸入預先定義的規則」，第 80 頁。

建立規則

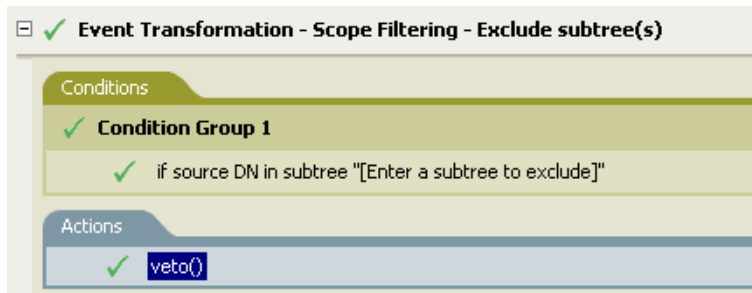
- 1 從「大綱」檢視窗或「規則流程」檢視窗中，選取「發行者」或「訂閱者」通道。
- 2 在「規則集」檢視窗中選取「事件轉換」規則集，然後按一下「建立或新增新規則至規則集」圖示 ，以建立新規則。
- 3 按一下「建立新規則」，然後再按「下一步」。
- 4 為規則命名。
- 5 使用填入的位置來將規則放入驅動程式內。

- 6 選取「建立規則後開啓編輯器」，然後再按「下一步」。
- 7 選取「DirXML 指令檔」為規則類型，然後按一下「完成」。
- 8 此時會出現檔案衝突視窗，其中內含「" 在編輯此項目前必須先儲存。是否要儲存編輯器的變更並繼續？"」訊息。請按一下「是」。此時會啟動「規則產生器」，並且會儲存新的「事件轉換」規則。

輸入預先定義的規則

- 1 在「規則產生器」中按一下滑鼠右鍵，然後按一下「新增」>「預先定義的規則」。
- 2 選取「事件轉換：範圍過濾：排除子網路樹」，然後按一下「確定」。
- 3 連按兩下「條件」索引標籤，以編輯條件。
- 4 刪除「值」欄位內的「[輸入要排除的子網路樹]」。
- 5 按一下瀏覽按鈕，瀏覽想要從同步化排除之事件的部份 Identity Vault 網路樹，然後按一下「確定」。
- 6 按一下「確定」。

7 按一下「檔案」>「儲存」以儲存規則。



規則的運作方式


如果您希望在同步化時，排除部份 Identity Vault，可以使用此規則。它可讓您不需使用「過濾器」的情形下同步化某些物件而排除其他物件。每次事件發生於 Identity Vault 之該特定部份時，該事件將會被否決。

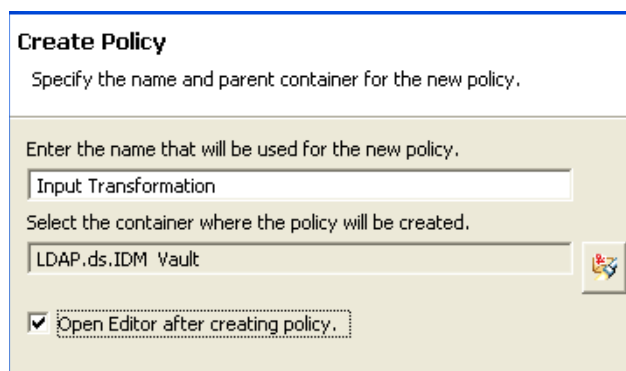
輸入或輸出轉換：將電話號碼從 (nnn) nnn-nnnn 重新格式化為 nnn-xxx-xxxx

在符合所需條件時，轉換電話號碼的格式。在驅動程式內的「輸入或輸出轉換」規則 (Policy) 上實作規則 (Rule)。您可以在「訂閱者」或「發行者」其中一個通道上實作規則，或同時在兩個通道上實作規則。

使用預先定義的規則包含兩個步驟：在「輸入或輸出轉換」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「輸入或輸出轉換」規則 (Policy)，請跳至「輸入預先定義的規則」，第 82 頁。

建立規則

- 1 從「大綱」檢視窗或「規則流程」檢視窗中，選取「發行者」或「訂閱者」通道。
- 2 在「規則集」檢視窗中選取「輸入或輸出轉換」規則集，然後按一下「建立或新增新規則至規則集」圖示 ，以建立新規則。
- 3 按一下「建立新規則」，然後再按「下一步」。
- 4 為規則命名。
- 5 使用填入的位置來將規則放入驅動程式內。

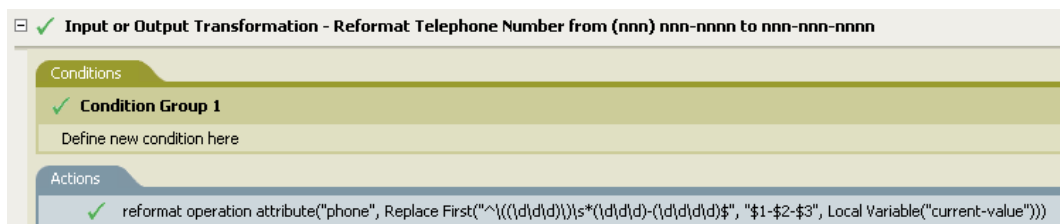


- 6 選取「建立規則後開啓編輯器」，然後再按「下一步」。
- 7 選取「DirXML 指令檔」為規則類型，然後按一下「完成」。

- 此時會出現檔案衝突視窗，其中內含「在編輯此項目前必須先儲存。是否要儲存編輯器的變更並繼續？」訊息。請按一下「是」。此時會啟動「規則產生器」，並且會儲存新的「輸入或輸出轉換」規則。

輸入預先定義的規則

- 在「規則產生器」內按一下滑鼠右鍵，然後按一下「新增」>「預先定義的規則」>「在之前插入預先定義的規則」或「在之後插入預先定義的規則」。
- 選取「輸入或輸出轉換：將「電話號碼」從 (nnn) nnn-nnnn 重新格式化為 nnn-*nnn-nnnn*」，然後按一下「確定」。
- 連按兩下「條件」索引標籤，以編輯條件。
- 定義要執行重新格式化電話號碼的條件。
- 按一下「確定」。
- 按一下「檔案」>「儲存」以儲存規則。



規則的運作方式


當您想要重新格式化電話號碼時，可以使用此規則。您必須定義符合要執行重新格式化電話號碼的條件。

輸入或輸出轉換：將電話號碼從 *nnn-nnn-nnnn* 重新格式化為 (nnn) *nnn-nnnn*

在符合所需條件時，轉換電話號碼的格式。在「輸入或輸出轉換」規則 (Policy) 上實作規則 (Rule)。您可以在「訂閱者」或「發行者」其中一個通道上實作規則，或同時在兩個通道上實作規則。

使用預先定義的規則包含兩個步驟；在「輸入或輸出轉換」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「輸入或輸出轉換」規則 (Policy)，請跳至「[輸入預先定義的規則](#)」，第 83 頁。

建立規則

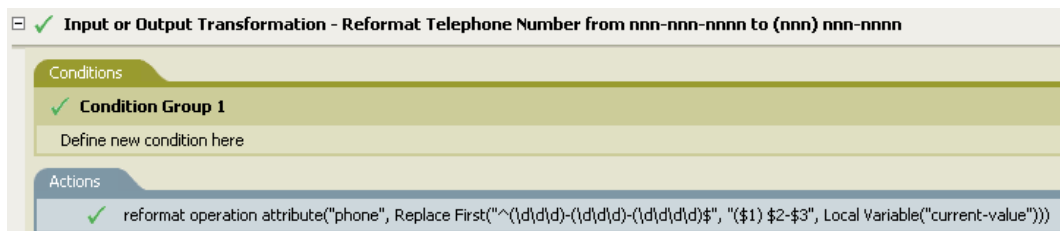
- 從「大綱」檢視窗或「規則流程」檢視窗中，選取「發行者」或「訂閱者」通道。
- 在「規則集」檢視窗中選取「輸入或輸出轉換」規則集，然後按一下「建立或新增新規則至規則集」圖示 ，以建立新規則。
- 按一下「建立新規則」，然後再按「下一步」。
- 為規則命名。

- 5 使用填入的位置來將規則放入驅動程式內。

- 6 選取「建立規則後開啓編輯器」，然後再按「下一步」。
- 7 選取「DirXML 指令檔」為規則類型，然後按一下「完成」。
- 8 此時會出現檔案衝突視窗，其中內含「在編輯此項目前必須先儲存。是否要儲存編輯器的變更並繼續？」訊息。請按一下「是」。此時會啓動「規則產生器」，並且會儲存新的「輸入或輸出轉換」規則。

輸入預先定義的規則

- 1 在「規則產生器」內按一下滑鼠右鍵，並按一下「新增」>「預先定義的規則」>在之前插入預先定義的規則」或「在之後插入預先定義的規則」。
- 2 按一下「輸入或輸出轉換：將「電話號碼」從 *nnn-**nnn**-**nnnn*** 重新格式化為 *(**nnn**) **nnn**-**nnnn***」，然後按一下「確定」。
- 3 連按兩下「條件」索引標籤，以編輯條件。
- 4 定義要執行重新格式化電話號碼的條件。
- 5 按一下「確定」。
- 6 按一下「檔案」>「儲存」以儲存規則。



規則的運作方式


當您想要重新格式化電話號碼時，可以使用此規則。您必須定義符合要執行重新格式化電話號碼的條件。

比對：鏡像複製發行者

使用指定點之資料儲存內的鏡像複製結構，比對 Identity Vault 內的物件。在驅動程式內的「相符」規則 (Policy) 上實作規則 (Rule)。您只可以在「發行者」通道上實作規則。

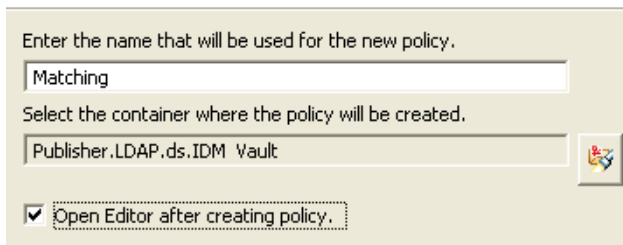
使用預先定義的規則包含兩個步驟：在「相符」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「相符」規則 (Policy)，請跳至[輸入預先定義的規則 \(第 84 頁\)](#)。

建立規則

- 1 從「大綱」檢視窗或「規則流程」檢視窗中，選取「發行者」通道。
- 2 在「規則集」檢視窗中選取「相符」規則集，然後按一下「建立或新增新規則至規則集」圖示 ，以建立新規則。
- 3 按一下「建立新規則」，然後再按「下一步」。
- 4 為規則命名。
- 5 使用填入的位置來將規則放入驅動程式內。


Create Policy

Specify the name and parent container for the new policy.

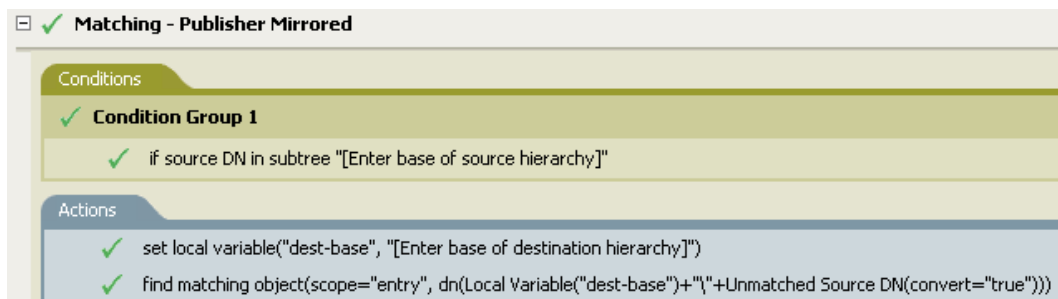


- 6 選取「建立規則後開啓編輯器」，然後再按「下一步」。
- 7 選取「DirXML 指令檔」為規則類型，然後按一下「完成」。
- 8 此時會出現檔案衝突視窗，其中內含「" 在編輯此項目前必須先儲存。是否要儲存編輯器的變更並繼續？"」訊息。請按一下「是」。此時會啓動「規則產生器」，並且會儲存新的「相符」規則。

輸入預先定義的規則

- 1 在「規則產生器」內按一下滑鼠右鍵，然後按一下「新增」>「預先定義的規則」>「在之前插入預先定義的規則」或「在之後插入預先定義的規則」。
- 2 選取「比對：鏡像複製發行者」，然後按一下「確定」。
- 3 連按兩下「條件」索引標籤，以編輯條件。
- 4 從「值」欄位中刪除 [輸入來源階層的基礎]。
- 5 在來源階層中瀏覽至要讓相符規則啓動的容器，並加以選取，然後按一下「確定」。
- 6 按一下「確定」。
- 7 連按兩下「動作」索引標籤，以編輯動作。
- 8 從「輸入字串」欄位中刪除 [輸入目的階層的基礎]。
- 9 按一下「編輯引數」圖示 ，以啓動「引數產生器」。
- 10 在「名詞」清單中選取「文字」。
- 11 連按兩下「文字」將其新增至引數中。
- 12 在編輯器中按一下瀏覽圖示，瀏覽至要與來源結構相符之目的階層內的容器，然後按一下「確定」。

- 13 按一下「確定」。
- 14 按一下「檔案」>「儲存」以儲存規則。



規則的運作方式


使用指定點之資料儲存內的鏡像複製結構，比對 Identity Vault 內的物件。當「新增」事件發生，且驅動程式檢查該物件是否存在時，此規則會開始檢查資料儲存內的指定 DN。接著驅動程式會設定一個 `dest-base` 的本地變數，做為 Identity Vault 內鏡像複製至資料儲存內結構的起始點。接著驅動程式會新增 `dest-base` 的本地變數，加上 \ 和物件的來源 DN，以建立其搜尋的網路位置。此規則會以斜線格式建立其尋找的路徑。

比對：鏡像複製訂閱者：**LDAP** 格式

使用指定點之 Identity Vault 內的鏡像複製結構，比對資料儲存內的物件。在驅動程式內的「相符」規則 (Policy) 上實作規則 (Rule)。您只可以在「訂閱者」通道上實作規則。

使用預先定義的規則包含兩個步驟：在「相符」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「相符」規則 (Policy)，請跳至「輸入預先定義的規則」，第 86 頁。

建立規則


- 1 從「大綱」檢視窗或「規則流程」檢視窗中，選取「發行者」通道。
- 2 在「規則集」檢視窗中選取「相符」規則集，然後按一下「建立或新增新規則至規則集」圖示 ，以建立新規則。
- 3 按一下「建立新規則」，然後再按「下一步」。
- 4 為規則命名。
- 5 使用填入的位置來將規則放入驅動程式內。

Create Policy

Specify the name and parent container for the new policy.

Enter the name that will be used for the new policy.


Select the container where the policy will be created.

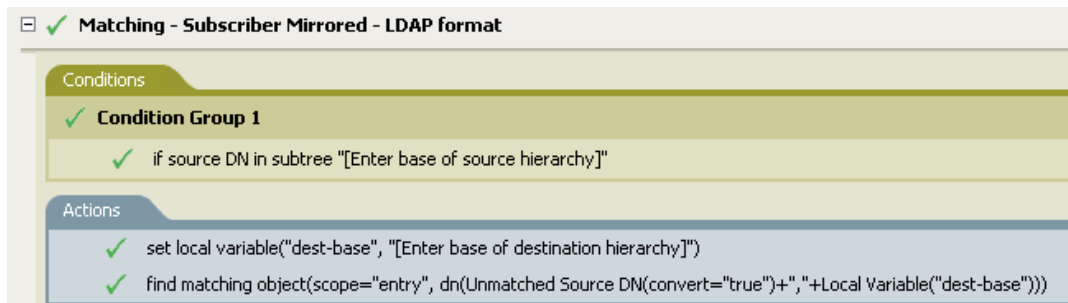
 

Open Editor after creating policy.

- 6 選取「建立規則後開啓編輯器」，然後再按「下一步」。
- 7 選取「DirXML 指令檔」為規則類型，然後按一下「完成」。
- 8 此時會出現檔案衝突視窗，其中內含「"在編輯此項目前必須先儲存。是否要儲存編輯器的變更並繼續？」訊息。請按一下「是」。此時會啓動「規則產生器」，並且會儲存新的「相符」規則。

輸入預先定義的規則

- 1 在「規則產生器」內按一下滑鼠右鍵，然後按一下「新增」>「預先定義的規則」>「在之前插入預先定義的規則」或「在之後插入預先定義的規則」。
- 2 選取「比對：鏡像複製訂閱者：LDAP 格式」，然後按一下「確定」。
- 3 連按兩下「條件」索引標籤，以編輯條件。
- 4 從「值」欄位中刪除 [輸入來源階層的基礎]。
- 5 在來源階層中瀏覽至要讓相符規則啓動的容器，並加以選取，然後按一下「確定」。
- 6 按一下「確定」。
- 7 連按兩下「動作」索引標籤，以編輯動作。
- 8 從「輸入字串」欄位中刪除 [輸入目的階層的基礎]。
- 9 按一下「編輯引數」圖示 ，以啓動「引數產生器」。
- 10 在「名詞」清單中選取「文字」。
- 11 連按兩下「文字」將其新增至引數中。
- 12 在編輯器中按一下瀏覽圖示，瀏覽至要與來源結構相符之目的階層內的容器，並加以選取，然後按一下「確定」。
- 13 按一下「確定」。
- 14 按一下「檔案」>「儲存」以儲存規則。



規則的運作方式


使用指定點之 Identity Vault 內的鏡像複製結構，比對資料儲存內的物件。當「新增」事件發生，且驅動程式檢查該物件是否存在時，此規則會開始檢查 Identity Vault 內的指定 DN。接著驅動程式會設定一個 dest-base 的本地變數，做為 Identity Vault 內鏡像複製至資料儲存內結構的起始點。接著驅動程式會新增物件的來源 DN 加上 a，以及 dest-base 的本地變數，以建立網路位置。此規則會建立其以 LDAP 格式尋找的路徑。

比對：依屬性值

依指定的屬性值比對物件。在驅動程式內的「相符」規則 (Policy) 上實作規則 (Rule)。您可以在「訂閱者」或「發行者」其中一個通道上實作規則，或同時在兩個通道上實作規則。

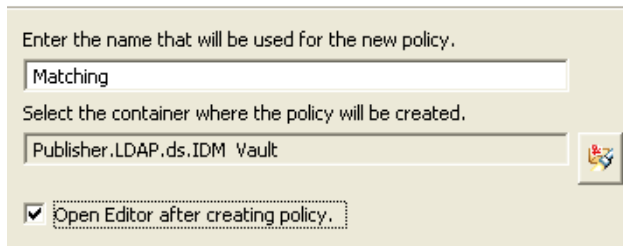
使用預先定義的規則包含兩個步驟；在「相符」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「相符」規則 (Policy)，請跳至「輸入預先定義的規則」，第 87 頁。

建立規則

- 1 從「大綱」檢視窗或「規則流程」檢視窗中，選取「發行者」通道。
- 2 在「規則集」檢視窗中選取「相符」規則集，然後按一下「建立或新增新規則至規則集」圖示 ，以建立新規則。
- 3 按一下「建立新規則」，然後再按「下一步」。
- 4 為規則命名。
- 5 使用填入的位置來將規則放入驅動程式內。

Create Policy

Specify the name and parent container for the new policy.



Enter the name that will be used for the new policy.

Matching


Select the container where the policy will be created.

Publisher.LDAP.ds.IDM Vault

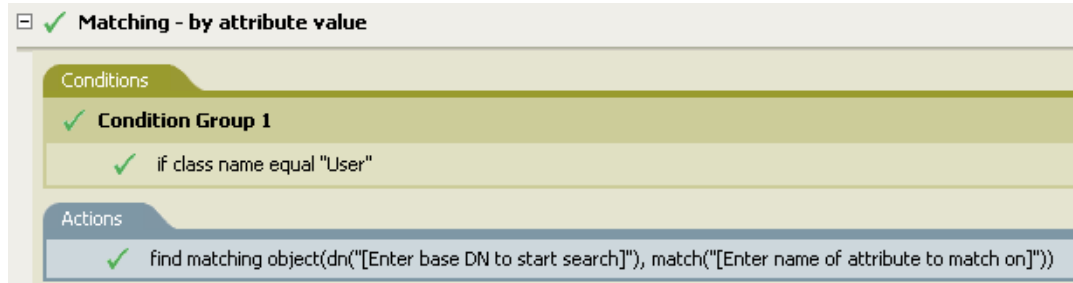
Open Editor after creating policy.

- 6 選取「建立規則後開啓編輯器」，然後再按「下一步」。
- 7 選取「DirXML 指令檔」為規則類型，然後按一下「完成」。
- 8 此時會出現檔案衝突視窗，其中內含「" 在編輯此項目前必須先儲存。是否要儲存編輯器的變更並繼續？"」訊息。請按一下「是」。此時會啟動「規則產生器」，並且會儲存新的「相符」規則。

輸入預先定義的規則

- 1 在「規則產生器」內按一下滑鼠右鍵，然後按一下「新增」>「預先定義的規則」>「在之前插入預先定義的規則」或「在之後插入預先定義的規則」。
- 2 選取「比對：依屬性值」，然後按一下「確定」。
- 3 連按兩下「動作」索引標籤，以編輯動作。
- 4 從「輸入 DN」欄位中刪除 / 輸入要開始搜尋的基礎 DN]。
- 5 按一下「編輯引數圖示」，以啟動「引數產生器」。
- 6 在「名詞」清單中選取「文字」。
- 7 連按兩下「文字」將其新增至引數中。
- 8 在「編輯器」中按一下瀏覽圖示，然後瀏覽至您希望搜尋啟動的容器並加以選取，然後按一下「確定」。

- 9 從「輸入相符屬性欄位」中刪除「[/ 輸入相符的屬性名稱/]」。
- 10 連接兩下「文字」將其新增至引數中。
- 11 按一下瀏覽圖示並選取希望與其相符的屬性。您可以選取一或多個與其相符的屬性，然後按一下「確定」。
- 12 按一下「確定」。
- 13 按一下「檔案」>「儲存」以儲存規則。



規則的運作方式


依指定的屬性值比對「使用者」物件。當「使用者」物件已同步化時，驅動程式會使用此規則來檢查並查看指定的屬性是否存在。若屬性不存在，則會建立新的「使用者」物件。

佈置：鏡像複製發行者

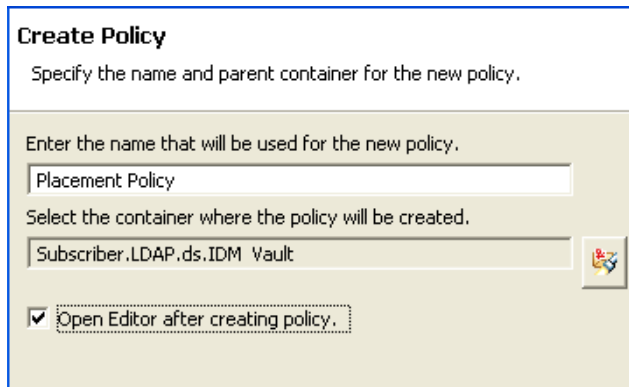
使用指定點之資料儲存內的鏡像複製結構，佈置 Identity Vault 內的物件。在驅動程式內的「佈置」規則 (Policy) 上實作規則 (Rule)。您只可以在「發行者」通道上實作規則。

使用預先定義的規則包含兩個步驟：在「佈置」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「佈置」規則 (Policy)，請跳至「輸入預先定義的規則」，第 89 頁。

建立規則


- 1 從「大綱」檢視窗或「規則流程」檢視窗中，選取「發行者」通道。
- 2 在規則集內選取「佈置」規則集，然後按一下「建立或新增新規則至規則集」圖示 ，以建立新規則。
- 3 按一下「建立新規則」，然後再按「下一步」。
- 4 為規則命名。

- 5 使用填入的位置來將規則放入驅動程式內。

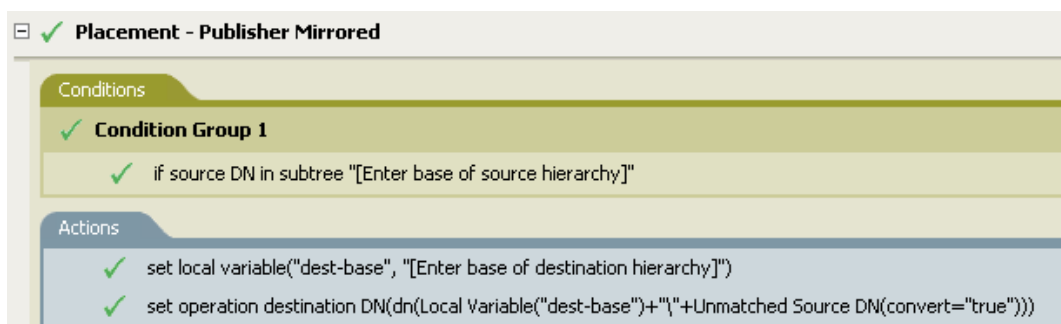


- 6 選取「建立規則後開啓編輯器」，然後再按「下一步」。
- 7 選取「DirXML 指令檔」為規則類型，然後按一下「完成」。
- 8 此時會出現檔案衝突視窗，其中內含「"在編輯此項目前必須先儲存。是否要儲存編輯器的變更並繼續?"」訊息。請按一下「是」。此時會啓動「規則產生器」，並且會儲存新的「佈置」規則。

輸入預先定義的規則

- 1 在「規則產生器」內按一下滑鼠右鍵，然後按一下「新增」>「預先定義的規則」>「在之前插入預先定義的規則」或「在之後插入預先定義的規則」。
- 2 選取「佈置：鏡像複製發行者」，然後按一下「確定」。
- 3 連按兩下「條件」索引標籤，以編輯條件。
- 4 從「值」欄位中刪除 [輸入來源階層的基礎]。
- 5 在來源階層中瀏覽至要讓物件採取動作的容器並加以選取，然後按一下「確定」。
- 6 連按兩下「動作」索引標籤，以編輯動作。
- 7 從「輸入字串」欄位中刪除 [輸入目的階層的基礎]。
- 8 按一下「編輯引數」圖示 ，以啓動「引數產生器」。
- 9 在「名詞」清單中選取「文字」。
- 10 連按兩下「文字」將其新增至引數中。
- 11 在「編輯器」中按一下瀏覽圖示，瀏覽至您希望放置物件之目的階層內的容器並加以選取，然後按一下「確定」。
- 12 按一下「確定」。

13 按一下「檔案」>「儲存」以儲存規則。



規則的運作方式


若「使用者」物件位於來源階層中，則物件會被放置在來自資料儲存的鏡像複製結構中。佈置規則會在本地變數 `dest-base` 定義時啟動。此規則會將「使用者」物件放置在 `dest-base` 不符的來源 DN 的位置中。此規則採用斜線格式。

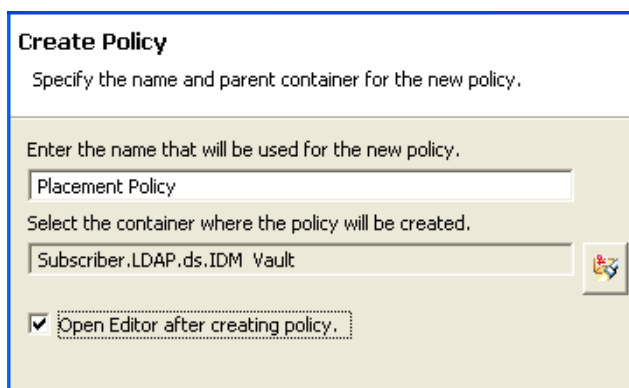
佈置：鏡像複製訂閱者：LDAP 格式

使用指定點之 Identity Vault 內的鏡像複製結構，將物件放置在資料儲存中。在驅動程式內的「佈置」規則 (Policy) 上實作規則 (Rule)。您只可以在「訂閱者」通道上實作規則。

使用預先定義的規則包含兩個步驟：在「佈置」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「佈置」規則 (Policy)，請跳至「輸入預先定義的規則」，第 91 頁。

建立規則


- 1 從「大綱」檢視窗或「規則流程」檢視窗中，選取「發行者」通道。
- 2 在「規則集」檢視窗內選取「佈置」規則集，然後按一下「建立或新增新規則至規則集」圖示 ，以建立新規則。
- 3 按一下「建立新規則」，然後再按「下一步」。
- 4 為規則命名。
- 5 使用填入的位置來將規則放入驅動程式內。

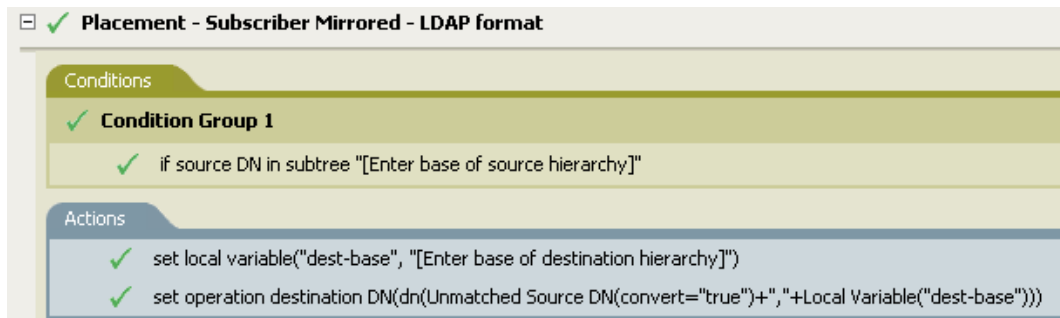


- 6 選取「建立規則後開啓編輯器」，然後再按「下一步」。
- 7 選取「DirXML 指令檔」為規則類型，然後按一下「完成」。

- 此時會出現檔案衝突視窗，其中內含「"在編輯此項目前必須先儲存。是否要儲存編輯器的變更並繼續？」訊息。請按一下「是」。此時會啟動「規則產生器」，並且會儲存新的「佈置」規則。

輸入預先定義的規則

- 在「規則產生器」內按一下滑鼠右鍵，然後按一下「新增」>「預先定義的規則」>「在之前插入預先定義的規則」或「在之後插入預先定義的規則」。
- 選取「佈置：鏡像複製訂閱者：LDAP 格式」，然後按一下「確定」。
- 連按兩下「條件」索引標籤，以編輯條件。
- 從「值」欄位中刪除 [輸入來源階層的基礎]。
- 在來源階層中瀏覽至要讓物件採取動作的容器，然後按一下「確定」。
- 連按兩下「動作」索引標籤，以編輯動作。
- 從「輸入字串」欄位中刪除 [輸入目的階層的基礎]。
- 按一下「編輯引數」圖示 ，以啟動「引數產生器」。
- 在「名詞」清單中選取「文字」。
- 連按兩下「文字」將其新增至引數中。
- 在編輯器中按一下瀏覽圖示，並瀏覽至您希望放置物件之目的階層內的容器，然後按一下「確定」。
- 按一下「確定」。
- 按一下「檔案」>「儲存」以儲存規則。



規則的運作方式


若「使用者」物件位於來源階層中，則物件會被放置在來自 Identity Vault 的鏡像複製結構中。佈置規則會在本地變數 `dest-base` 定義時啟動。此規則會將「使用者」物件放置在不符合的來源 DN 的位置，`dest-base` 中。此規則採用 LDAP 格式。

佈置：一般發行者

將來自資料儲存的物件放置到 Identity Vault 內的某個容器中。在驅動程式內的「佈置」規則 (Policy) 上實作規則 (Rule)。您只可以在「發行者」通道上實作規則。


使用預先定義的規則包含兩個步驟：在「佈置」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「佈置」規則 (Policy)，請跳至「輸入預先定義的規則」，第 92 頁。

建立規則

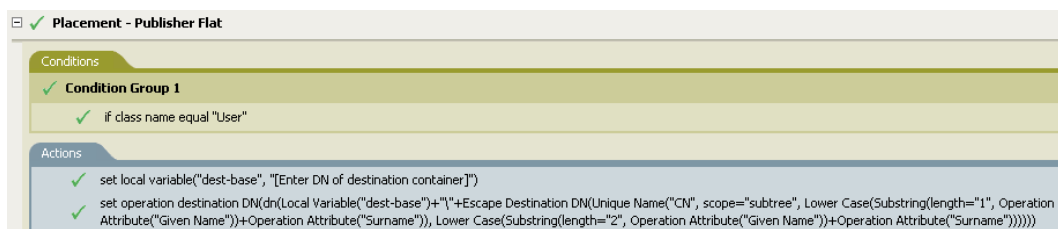
- 1 從「大綱」檢視窗或「規則流程」檢視窗中，選取「發行者」通道。
- 2 在「規則集」檢視窗內選取「佈置」規則集，然後按一下「建立或新增新規則至規則集」圖示 ，以建立新規則。
- 3 按一下「建立新規則」，然後再按「下一步」。
- 4 為規則命名。
- 5 使用填入的位置來將規則放入驅動程式內。

- 6 選取「建立規則後開啓編輯器」，然後再按「下一步」。
- 7 選取「DirXML 指令檔」為規則類型，然後按一下「完成」。
- 8 此時會出現檔案衝突視窗，其中內含「" 在編輯此項目前必須先儲存。是否要儲存編輯器的變更並繼續？"」訊息。請按一下「是」。此時會啓動「規則產生器」，並且會儲存新的「佈置」規則。

輸入預先定義的規則

- 1 在「規則產生器」內按一下滑鼠右鍵，然後按一下「新增」>「預先定義的規則」>「在之前插入預先定義的規則」或「在之後插入預先定義的規則」。
- 2 選取「佈置：一般發行者」，然後按一下「確定」。
- 3 連按兩下「動作」索引標籤，以編輯動作。
- 4 從「輸入字串」欄位中刪除「[輸入目的容器的 DN]」。
- 5 按一下「編輯引數」圖示 ，以啓動「引數產生器」。
- 6 在「名詞」清單中選取「文字」。
- 7 連按兩下「文字」將其新增至引數中。
- 8 在「編輯器」中按一下瀏覽圖示，然後瀏覽至要放置所有「使用者」物件之目的容器並加以選取，然後按一下「確定」。
- 9 按一下「確定」。

10 按一下「檔案」>「儲存」以儲存規則。



規則的運作方式


此規則會將所有「使用者」物件放置到目的 DN 中。此規則會將目的容器的 DN 設定為本地變數 `dest-base`。此規則接著會將目的 DN 設定為 `dest-base\CN` 屬性。「使用者」物件的 CN 屬性值由 Given Name 屬性的前兩個字母加上 Surname 屬性的小寫字母組成。此規則採用斜線格式。

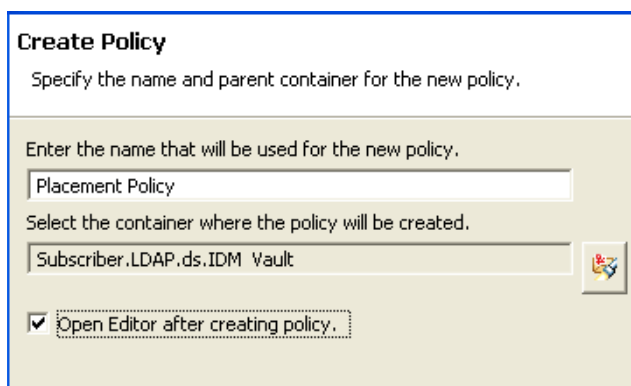
佈置：一般訂閱者：LDAP 格式

將來自 Identity Vault 的物件放置到資料儲存內的某個容器中。在驅動程式內的「訂閱者佈置」規則 (Policy) 上實作規則 (Rule)。

使用預先定義的規則包含兩個步驟：在「佈置」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「佈置」規則 (Policy)，請跳至「輸入預先定義的規則」，第 94 頁。

建立規則


- 1 從「大綱」檢視窗或「規則流程」檢視窗中，選取「發行者」通道。
- 2 在「規則集」檢視窗內選取「佈置」規則集，然後按一下「建立或新增新規則至規則集」圖示 ，以建立新規則。
- 3 按一下「建立新規則」，然後再按「下一步」。
- 4 為規則命名。
- 5 使用填入的位置來將規則放入驅動程式內。

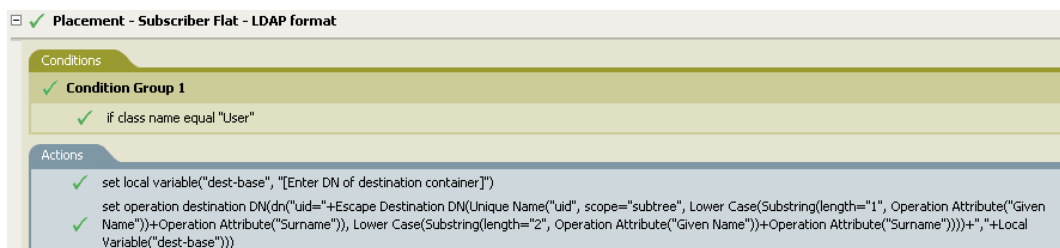


- 6 選取「建立規則後開啓編輯器」，然後再按「下一步」。
- 7 選取「DirXML 指令檔」為規則類型，然後按一下「完成」。

- 此時會出現檔案衝突視窗，其中內含「在編輯此項目前必須先儲存。是否要儲存編輯器的變更並繼續？」訊息。請按一下「是」。此時會啟動「規則產生器」，並且會儲存新的「佈置」規則。

輸入預先定義的規則

- 在「規則產生器」內按一下滑鼠右鍵，然後按一下「新增」>「預先定義的規則」>「在之前插入預先定義的規則」或「在之後插入預先定義的規則」。
- 選取「佈置：一般訂閱者：LDAP 格式」，然後按一下「確定」。
- 連按兩下「動作」索引標籤，以編輯動作。
- 從「輸入字串」欄位中刪除「[輸入目的容器的 DN]」。
- 按一下「編輯引數」圖示 ，以啟動「引數產生器」。
- 在「名詞」清單中選取「文字」。
- 連按兩下「文字」將其新增至引數中。
- 在編輯器中，新增您希望放置所有「使用者」物件的目的容器。請確定使用 LDAP 格式指定容器，然後按一下「確定」。
- 按一下「確定」。
- 按一下「檔案」>「儲存」以儲存規則。



規則的運作方式


此規則會將所有「使用者」物件放置到目的 DN 中。此規則會將目的容器的 DN 設定為本地變數 `dest-base`。此規則接著會將目的 DN 設定為 `uid=` 唯一名稱 `dest-base`。「使用者」物件的 `uid` 屬性值由 `Given Name` 屬性的前兩個字母加上 `Surname` 屬性的小寫字母組成。此規則採用 LDAP 格式。

佈置：依部門的發行者

將來自資料儲存內某容器的物件放置到 Identity Vault 內的多個容器中。在驅動程式內的「佈置」規則 (Policy) 上實作規則 (Rule)。您只可以在「發行者」通道上實作規則。

使用預先定義的規則包含兩個步驟：在「佈置」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「佈置」規則 (Policy)，請跳至「輸入預先定義的規則」，第 95 頁。


建立規則

- 從「大綱」檢視窗或「規則流程」檢視窗中，選取「發行者」通道。
- 在「規則集」檢視窗內選取「佈置」規則集，然後按一下「建立或新增新規則至規則集」圖示 ，以建立新規則。

- 3 按一下「建立新規則」，然後再按「下一步」。
- 4 為規則命名。
- 5 使用填入的位置來將規則放入驅動程式內。

- 6 選取「建立規則後開啓編輯器」，然後再按「下一步」。
- 7 選取「DirXML 指令檔」為規則類型，然後按一下「完成」。
- 8 此時會出現檔案衝突視窗，其中內含「在編輯此項目前必須先儲存。是否要儲存編輯器的變更並繼續？」訊息。請按一下「是」。此時會啟動「規則產生器」，並且會儲存新的「佈置」規則。

輸入預先定義的規則

- 1 在「規則產生器」內按一下滑鼠右鍵，然後按一下「新增」>「預先定義的規則」>「在之前插入預先定義的規則」或「在之後插入預先定義的規則」。
- 2 選取「佈置：依部門的發行者」，然後按一下「確定」。
- 3 連接兩下「動作」索引標籤，以編輯動作。
- 4 從「輸入字串欄位」中刪除「[輸入目的組織的 DN]」。
- 5 按一下「編輯引數」圖示 , 以啟動「引數產生器」。
- 6 在「名詞」清單中選取「文字」。
- 7 連接兩下「文字」將其新增至引數中。
- 8 在「編輯器」中按一下瀏覽圖示，然後瀏覽至 Identity Vault 內的父容器並加以選取。請確定所有的部門容器皆為此 DN 的子容器，然後按一下「確定」。
- 9 按一下「確定」。
- 10 按一下「檔案」>「儲存」以儲存規則。

規則的運作方式

此規則會依據儲存在 OU 屬性內的值，將「使用者」物件放置在適當的部門容器中。若需要放置「使用者」物件，並且其擁有可用的 OU 屬性，則「使用者」物件會被放置在 `dest-base\OU` 屬性的值 \CN 屬性中。

`dest-base` 為本地變數。DN 必須為相對於部門容器的根路徑。其可為組織或組織單位。儲存於 OU 屬性內的值必須為 `dest-base` 本地變數的子容器名稱。

子容器必須與要放置使用者物件之位置關聯。OU 屬性的值必須為子容器的名稱。如果 OU 屬性不存在，則不會執行此規則。


「使用者」物件的 CN 屬性值由 Given Name 屬性的前兩個字母加上 Surname 屬性的小寫字母組成。此規則採用斜線格式。

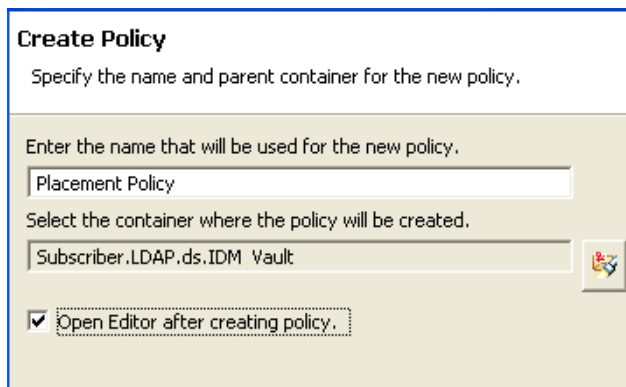
佈置：依部門的訂閱者：**LDAP** 格式

將來自 Identity Vault 內某容器的物件依 OU 屬性放置到資料儲存內的多個容器中。在驅動程式內的「佈置」規則 (Policy) 上實作規則 (Rule)。您只可以在「訂閱者」通道上實作規則。

使用預先定義的規則包含兩個步驟：在「佈置」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「佈置」規則 (Policy)，請跳至「[輸入預先定義的規則](#)」，第 97 頁。


建立規則

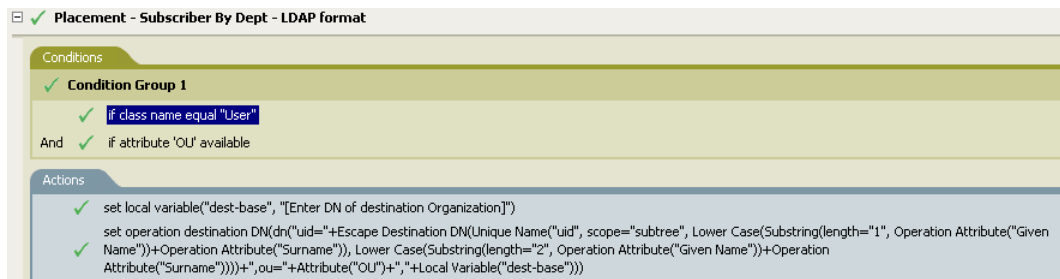
- 1 從「大綱」檢視窗或「規則流程」檢視窗中，選取「發行者」通道。
- 2 在「規則集」檢視窗內選取「佈置」規則集，然後按一下「建立或新增新規則至規則集」圖示 ，以建立新規則。
- 3 按一下「建立新規則」，然後再按「下一步」。
- 4 為規則命名。
- 5 使用填入的位置來將規則放入驅動程式內。



- 6 選取「建立規則後開啓編輯器」，然後再按「下一步」。
- 7 選取「DirXML 指令檔」為規則類型，然後按一下「完成」。
- 8 此時會出現檔案衝突視窗，其中內含「在編輯此項目前必須先儲存。是否要儲存編輯器的變更並繼續？」訊息。請按一下「是」。此時會啟動「規則產生器」，並且會儲存新的「佈置」規則。

輸入預先定義的規則

- 1 在「規則產生器」內按一下滑鼠右鍵，然後按一下「新增」>「預先定義的規則」>在之前插入預先定義的規則」或「在之後插入預先定義的規則」。
- 2 選取「佈置：依部門的訂閱者：LDAP 格式」，然後按一下「確定」。
- 3 連按兩下「動作」索引標籤，以編輯動作。
- 4 從「輸入字串」欄位中刪除「[輸入目的組織的 DN]」。
- 5 按一下「編輯引數」圖示 ，以啟動「引數產生器」。
- 6 在「名詞」清單中選取「文字」。
- 7 連按兩下「文字」將其新增至引數中。
- 8 在「編輯器」中新增資料儲存內的父容器。父容器必須以 LDAP 格式指定。請確定所有的部門容器皆為此 DN 的子容器，然後按一下「確定」。
- 9 按一下「確定」。
- 10 按一下「檔案」>「儲存」以儲存規則。



規則的運作方式

此規則會依據儲存在 OU 屬性內的值，將「使用者」物件放置在適當的部門容器中。若需要放置「使用者」物件，並且其擁有可用的 OU 屬性，則「使用者」物件會被放置在 uid=唯一名稱,ou=OU 屬性的值,dest-base 中。

dest-base 為本地變數。DN 必須為相對於部門容器的根路徑。其可為組織或組織單位。儲存於 OU 屬性內的值必須為 dest-base 本地變數的子容器名稱。

子容器必須與要放置「使用者」物件之位置關聯。OU 屬性的值必須為子容器的名稱。如果 OU 屬性不存在，則不會執行此規則。

「使用者」物件的 uid 屬性值由 Given Name 屬性的前兩個字母加上 Surname 屬性的小寫字母組成。此規則採用 LDAP 格式。

2.2.7 使用「規則模擬器」測試規則

「規則模擬器」可讓您在驅動程式流程的任意點上執行規則，並且不需在 Identity Vault 內實作規則即可查看結果。您可以在不影響生產環境或已連接系統的情形下測試規則。

如需使用「規則模擬器」之常用工作的相關資訊，請參閱以下各節：

- 「存取「規則模擬器」」，第 98 頁
- 「使用「規則模擬器」」，第 99 頁

「規則模擬器」使用 XML。eDirectory 文件類型定義檔 (nds.dtd) 定義了 Metadirectory 引擎能夠處理的 XML 文件綱要。不符合此綱要的 XML 文件會產生錯誤。若要驗證文件是否符合 nds.dtd，並尋找與錯誤發生原因相關的資訊，請參閱「eDirectory DTD 的指令和事件 (<http://developer.novell.com/ndk/doc/dirxml/index.html?page=/ndk/doc/dirxml/dirxmlbk/data/a36pjzu.html>)」。


「規則模擬器」無法從應用程式驅動程式 (例如：SOAP 和分隔文字) 中模擬啓始的規則集。這些驅動程式會使用以逗號分隔的檔案或文字檔做為輸入，而 XML 或 XDS 則衍生自規則鏈內的規則。目前「規則模擬器」僅接受有效的 XML 或 XDS 做為輸入值。其他的功能將考慮在未來的發行版本中提供。

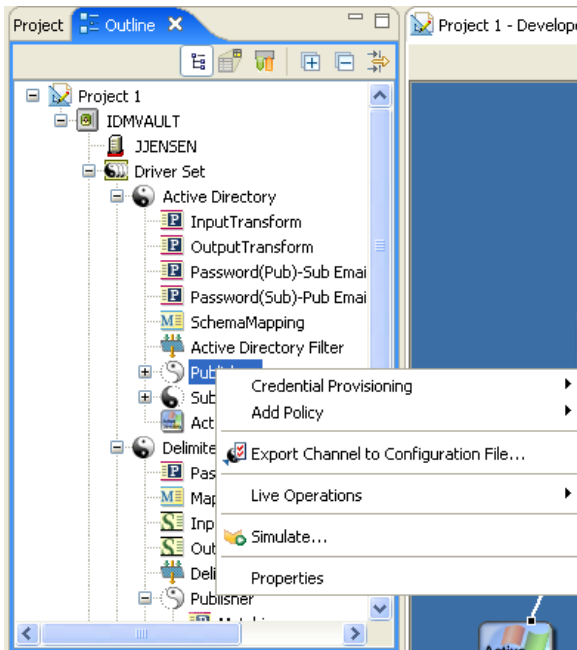
存取「規則模擬器」

「規則模擬器」可以透過三種方式來存取：


- ◆ 「大綱檢視」，第 98 頁
- ◆ 「規則流程」，第 98 頁
- ◆ 「編輯器」，第 99 頁

大綱檢視

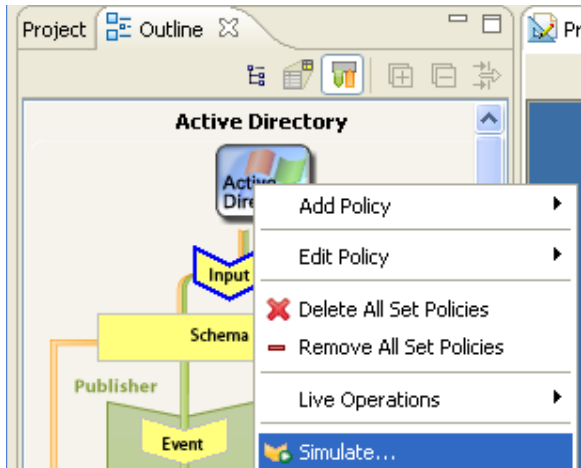
- 1 按一下「顯示模型大綱」圖示 。
- 2 在驅動程式、發行者、訂閱者、映射規則、過濾器或任何您想要模擬的規則上按一下滑鼠右鍵，然後按一下「模擬」。




規則流程

- 1 按一下「顯示規則流程」圖示 。

- 2 在輸入、輸出、schemaMapping、過濾器 and 任何您想要模擬的規則集上按一下滑鼠右鍵，然後按一下「模擬」。



編輯器

您可以透過選取「規則產生器」、「綱要映射」編輯器或「過濾器」編輯器工具列內的「規則模擬器圖示」，存取「規則模擬器」。

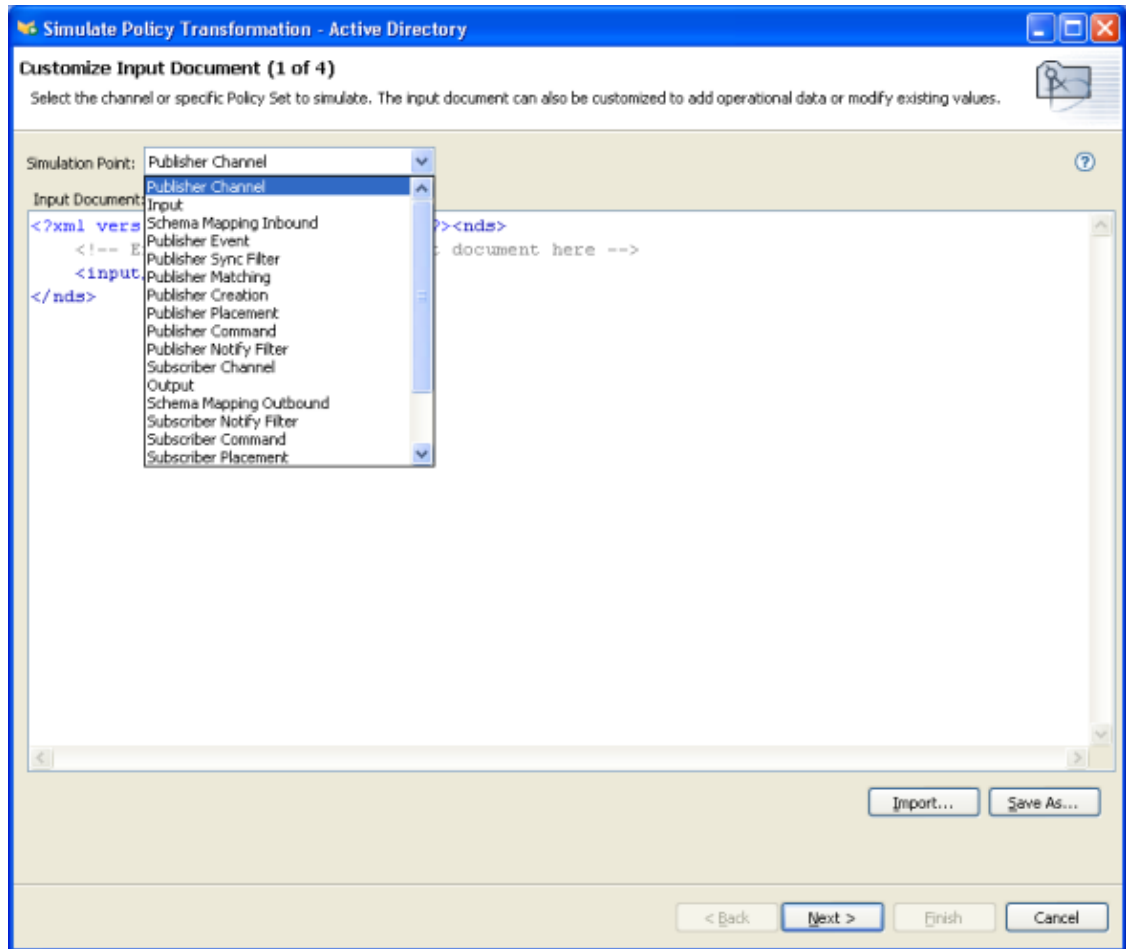
使用「規則模擬器」

「規則模擬器」可讓您選取驅動程式流程內的某一點，以使用特定操作來測試規則。它可讓您在測試時編輯輸入和輸出文件。若您想要保留變更，請選取「另存新檔」圖示，將文件儲存為 XML 檔案。

若要使用「規則模擬器」，請執行下列動作：

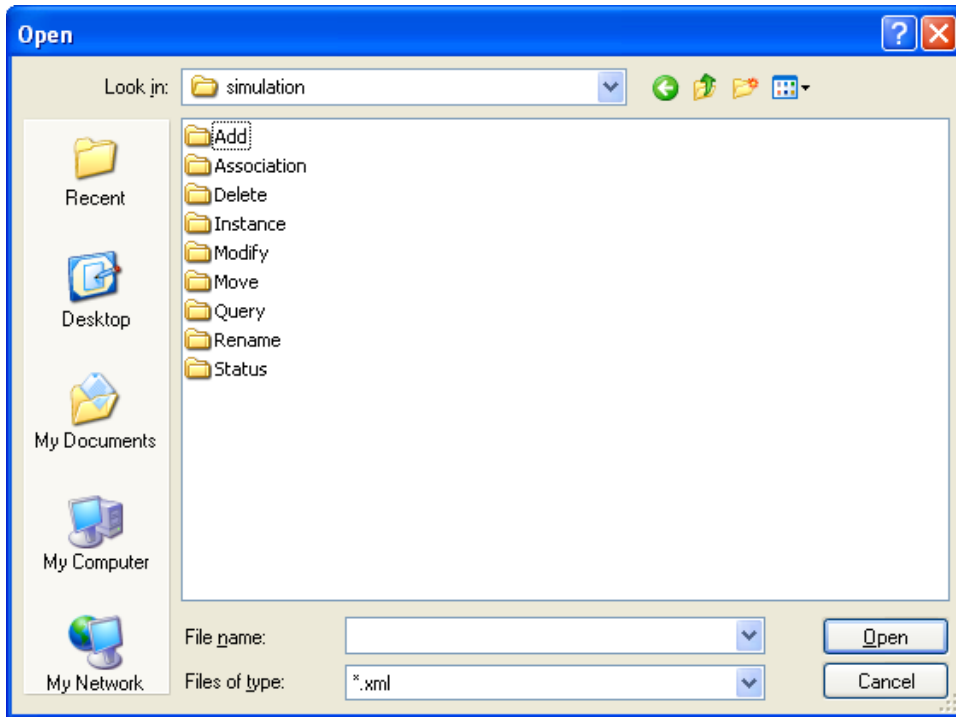
- 1 從「模擬點」下拉式清單中選取驅動程式流程內要測試規則的位置。您可以選取下列任何項目：發行者通道、訂閱者通道、輸入、綱要映射、事件、同步化過濾器、相符、建立、佈置、指令和通知過濾器。

若您選取特定規則 (Policy) 或規則 (Rule) 以進行測試，「模擬點」選項僅會顯示「目的 NDS」或「來源 NDS」。

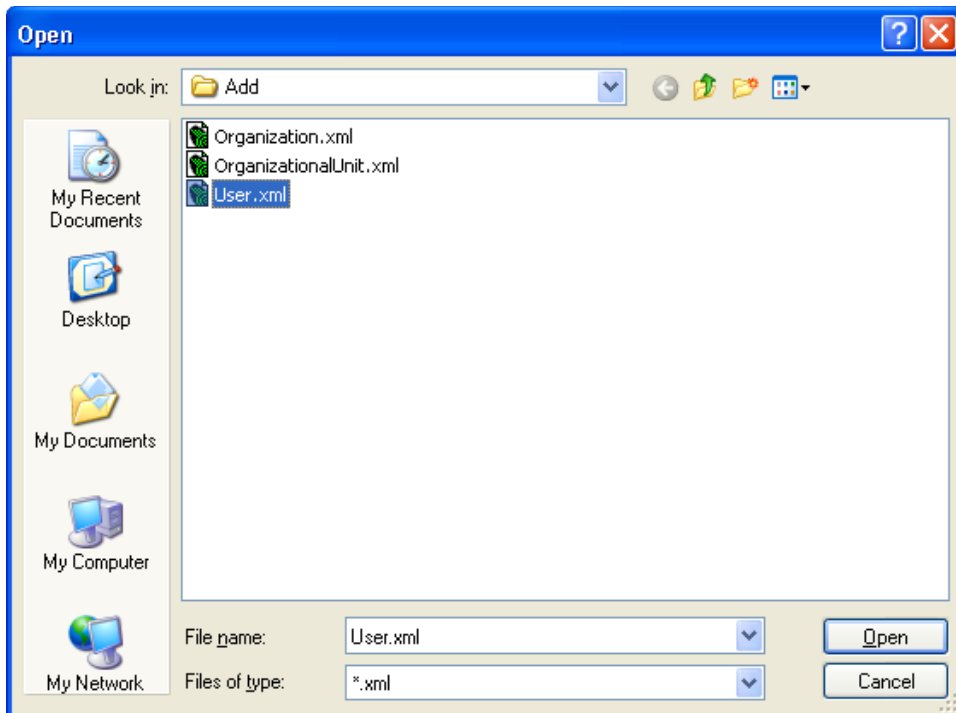


2 選取「輸入」，然後瀏覽至要測試的檔案並加以選取。

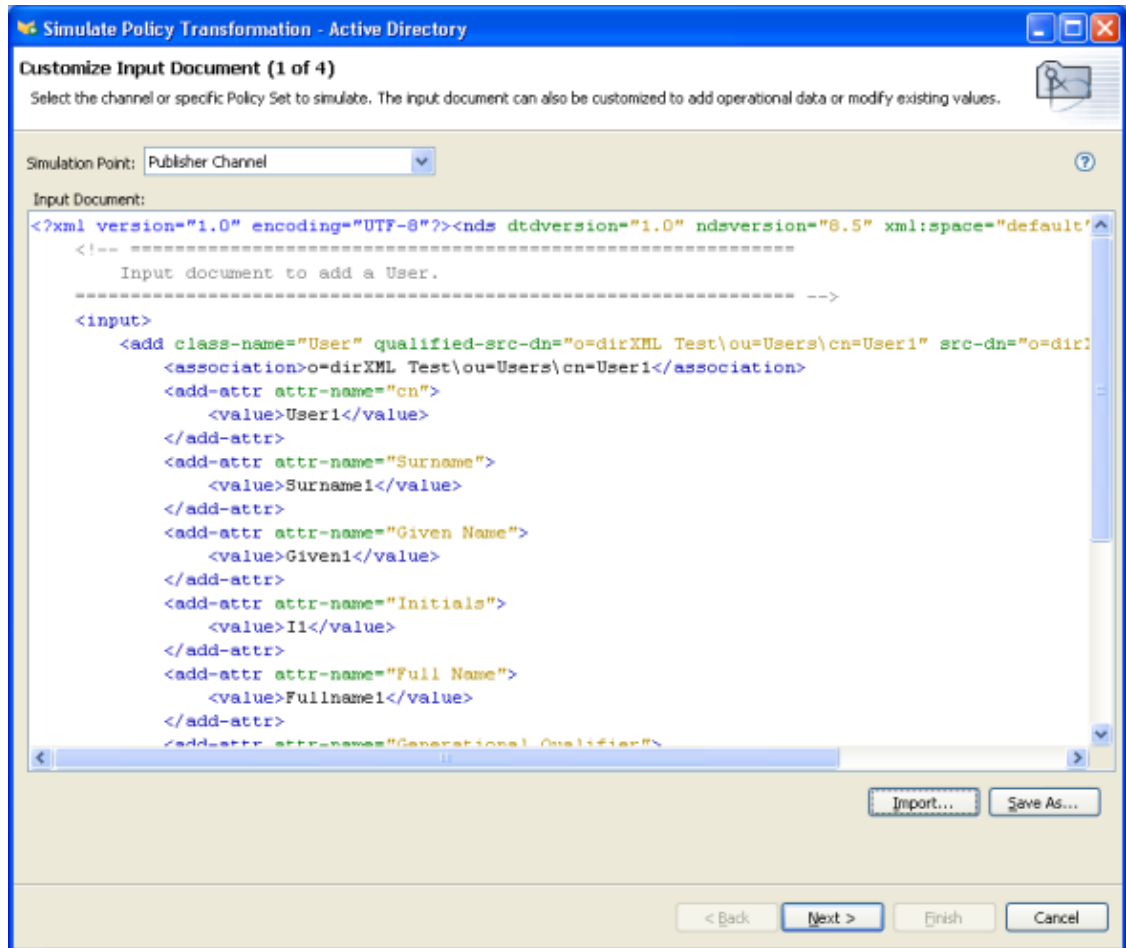
Designer 隨附了範例事件檔，可供您使用。檔案位於外掛程式 `com.novell.designer.idm.policy.simulation`。其事件為「新增」、「關聯」、「刪除例項」、「修改」、「移動」、「查詢」、「重新命名」和「狀態」。



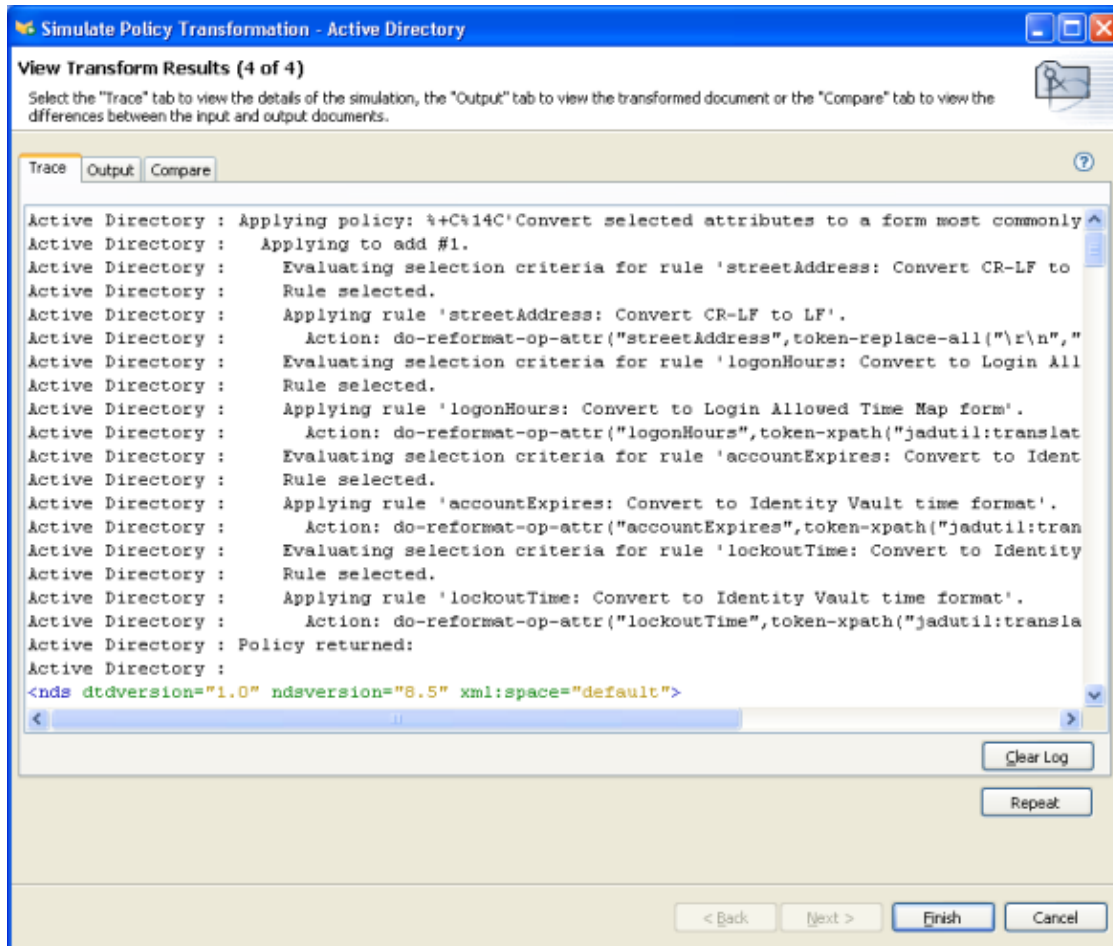
- 3 連按兩下資料夾以顯示可用的事件。您可於每個物件選取不同的檔案。例如，若您選取「Add」，則有三個選項：Organization.xml、OrganizationalUnit.xml 和 User.xml。檔案會指出事件。若您選取 User.xml，其為使用者物件的「新增」事件。



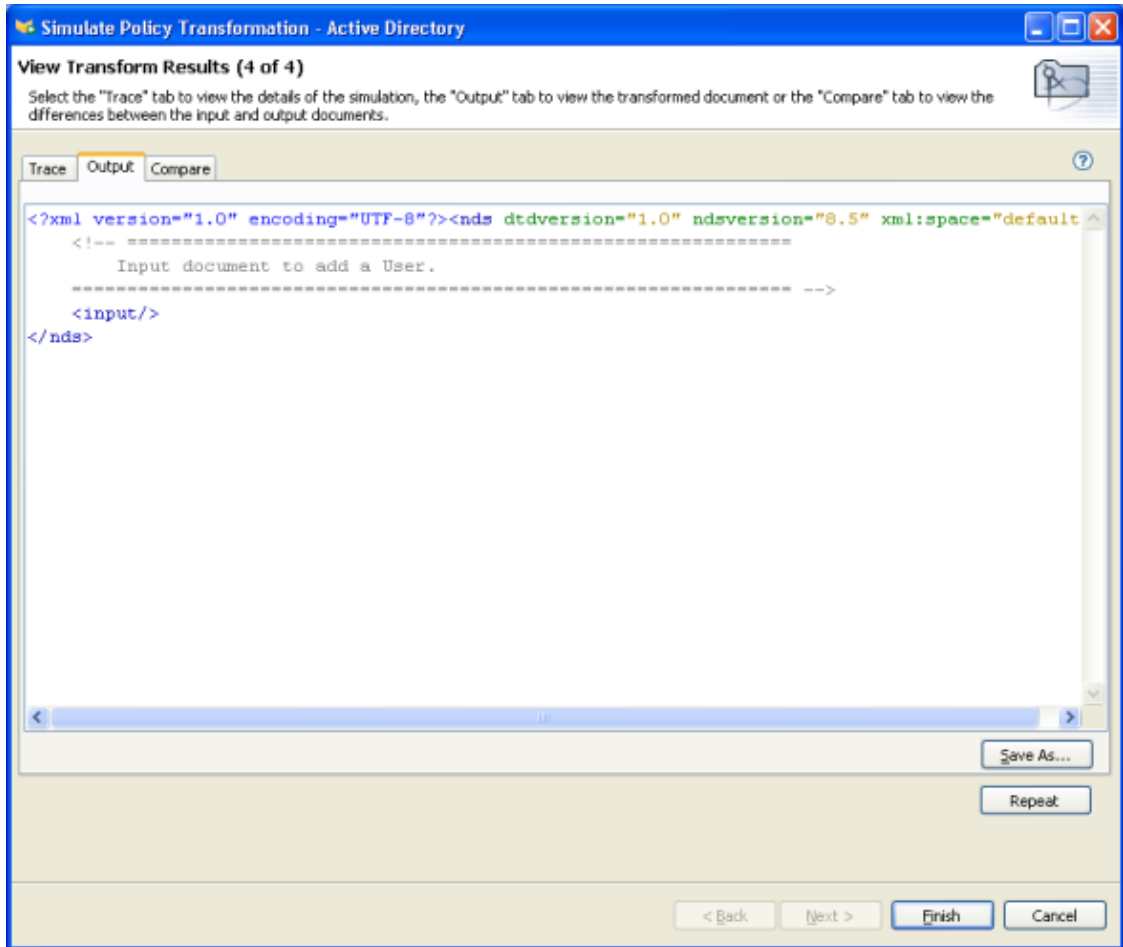
- 4 選取某個檔案，然後按一下「開啓」，在視窗內顯示輸入文件。
- 5 按「下一步」。



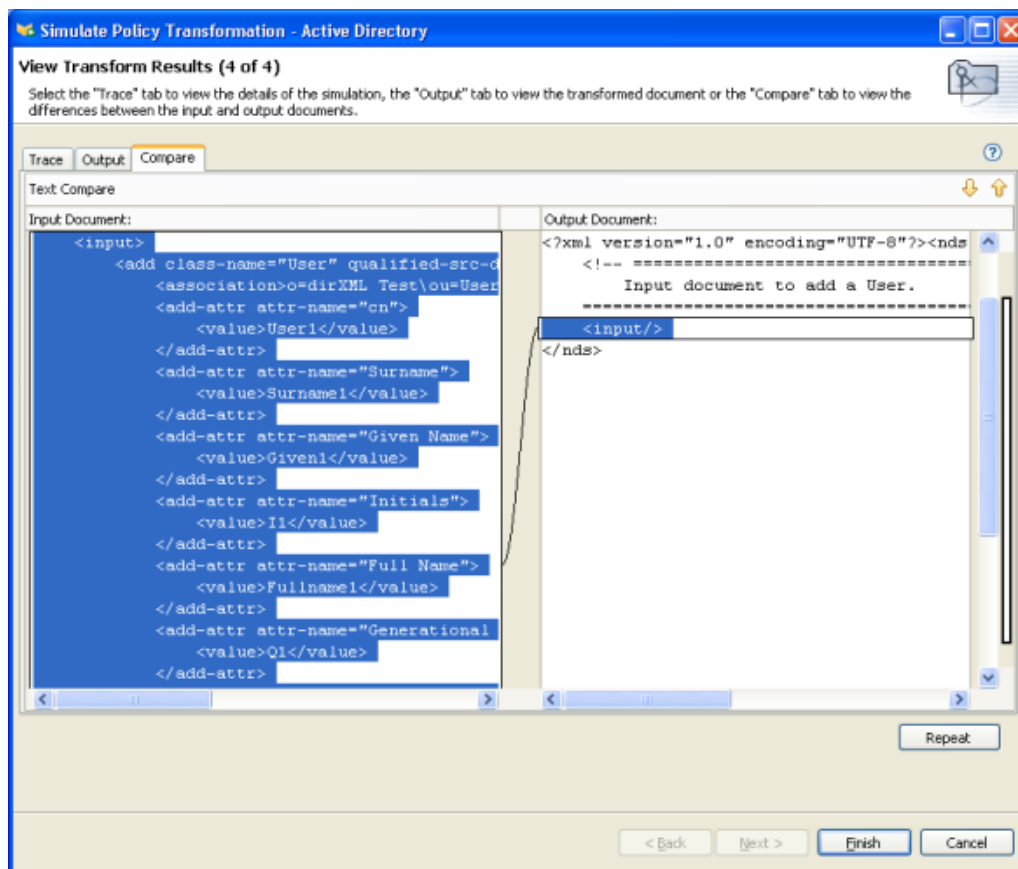
- 6 選取「追蹤」索引標籤以查看已處理規則之事件的結果。此視窗內的資訊與您在 DSTRACE 內所見的資訊相同。



7 選取「輸出」索引標籤，以查看已產生的輸出文件。



8 選取「比較」索引標籤，以比較輸出文件與輸入文件。



9 當您結束查看結果時，請按一下「重複」，以針對規則測試另一個事件。

10 當您完成測試時，請按一下「完成」，關閉「規則模擬器」。

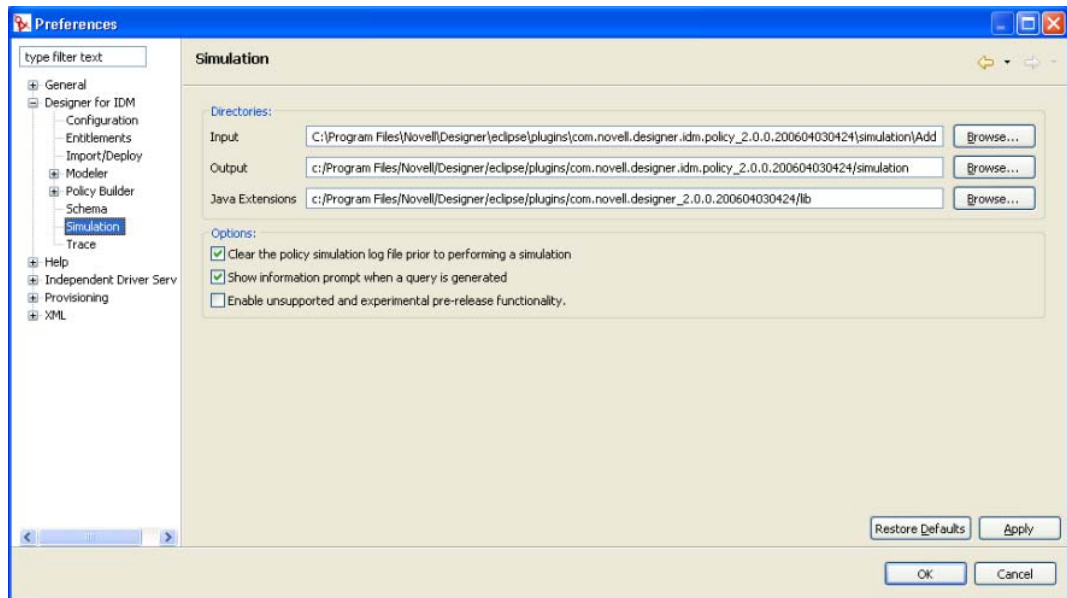
使用 Java 延伸功能模擬規則

包含參考至外部 Java 延伸功能的規則，現在可以透過指定 jar 檔案的目錄位置進行模擬。

若要判定或變更延伸功能的目錄，請執行下列動作：

- 1 從工具列選取「Windows > 優先設定」。
- 2 瀏覽至「Designer for IDM > 模擬」頁面。

3 將包含 Java 類別的 jar 檔複製到指定目錄，並模擬規則。



附註：「啓用未支援及實驗性的發行前功能」選項可讓「規則模擬器」針對即時的 Identity Vault 或已連接系統測試規則。此選項在 Designer 1.2 並未支援，並且未提供文件。

2.2.8 編輯 DirXML 程序檔

Designer 可讓您藉由使用 XML 編輯器或文字編輯器來檢視、編輯及驗證 XML。

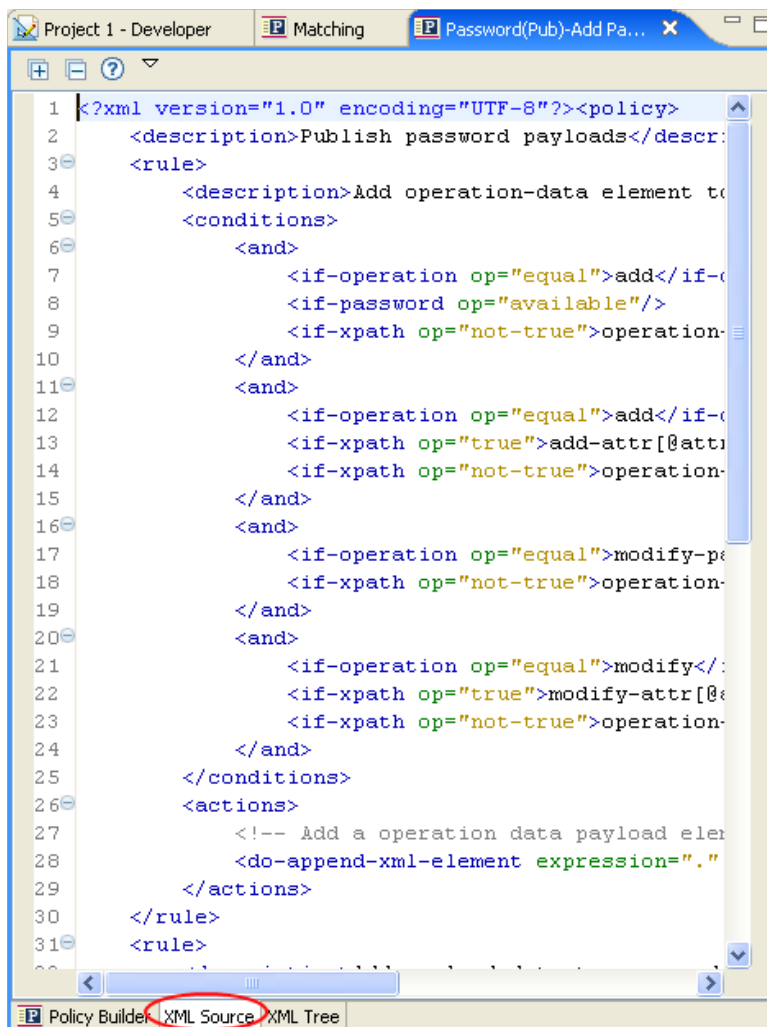
- ◆ 「檢視 XML 來源」，第 106 頁
- ◆ 「編輯 XML 來源」，第 110 頁
- ◆ 「驗證 XML 來源」，第 112 頁

檢視 XML 來源

您可以 XML 或以 XML 樹狀結構格式檢視「XML 來源」。

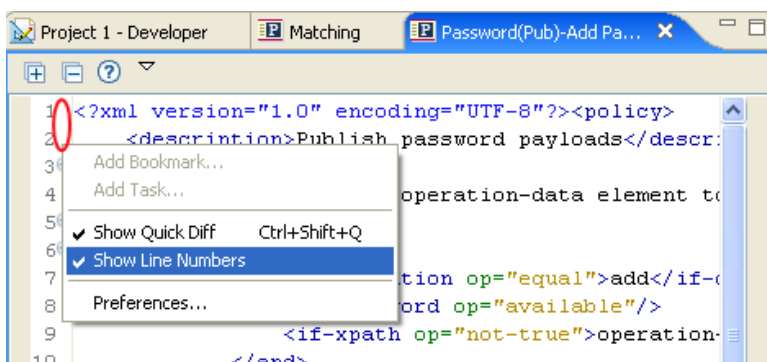
若要開啓 XML 來源檢視窗，請執行下列動作：

- 1 按一下位於「規則產生器」工作空間底部的「XML 來源」。



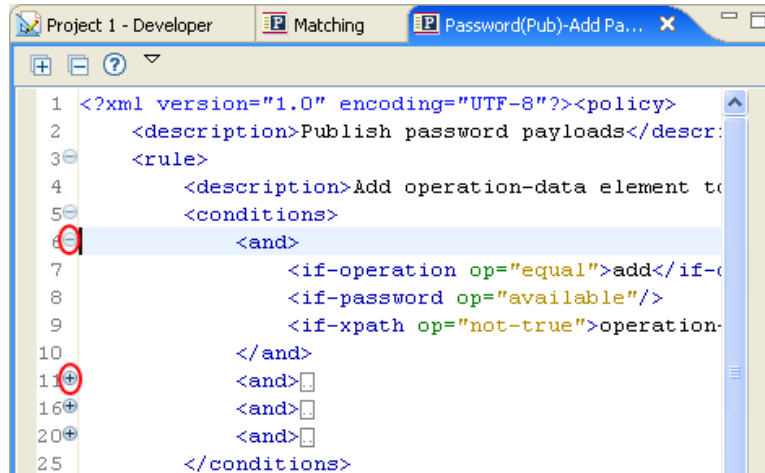
XML 編輯器會顯示行號。

- 2 若要查看行號，請在左邊界內按一下滑鼠右鍵，然後選取「顯示行號」。



XML 編輯器會依功能展開或折疊 XML。若其中包含大量 XML 的功能，您可以按一下位於左上角的減號圖示，以折疊 XML。

- 3 若要展開所有的 XML 功能，請按一下位於左邊角落的加號圖示。每個元素的左邊界皆擁有其各自的加號或減號圖示。



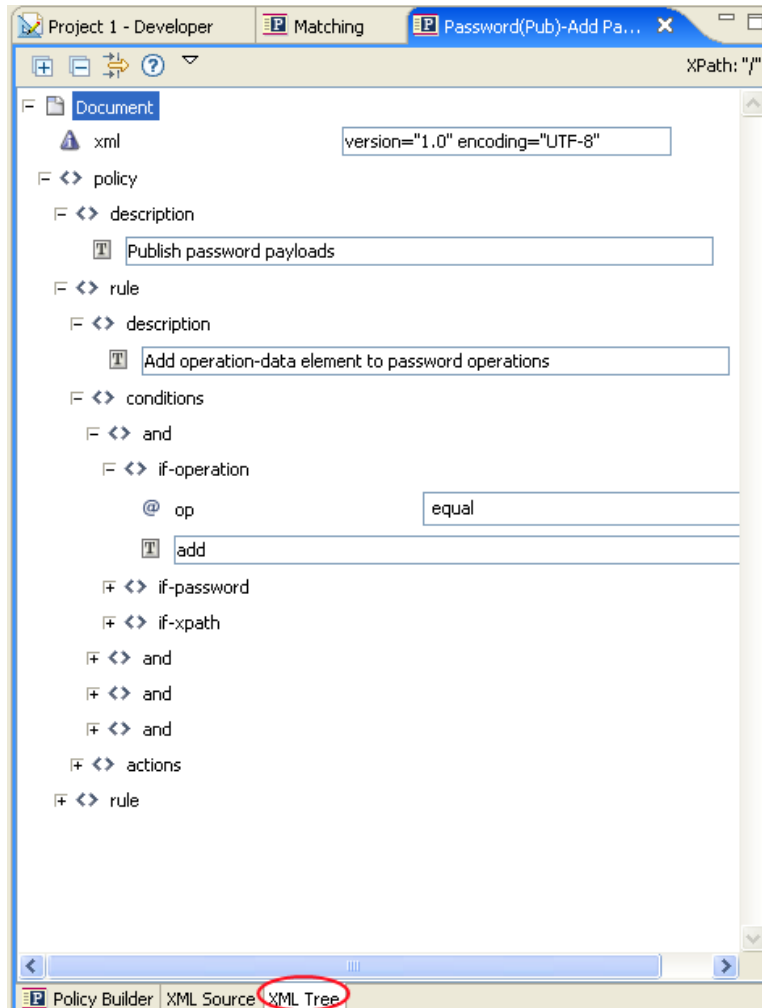
The screenshot shows an XML editor window titled "Project 1 - Developer" with a file named "Password(Pub)-Add Pa...". The XML content is as follows:

```
1 <?xml version="1.0" encoding="UTF-8"?><policy>
2   <description>Publish password payloads</description>
3   <rule>
4     <description>Add operation-data element to
5     <conditions>
6       <and>
7         <if-operation op="equal">add</if-operation>
8         <if-password op="available"/>
9         <if-xpath op="not-true">operation-
10      </and>
11     </and>
16    </and>
20    </and>
25  </conditions>
```

On the left side of the editor, there are expand/collapse icons for each element. A red circle highlights the minus sign icon next to line 6, indicating that the <and> element is currently collapsed. Other plus sign icons are visible next to lines 11, 16, and 20.

若要以 XML 樹狀結構格式檢視 XML：

- 1 按一下位於「規則產生器」工作空間底部的「XML 樹狀結構」。

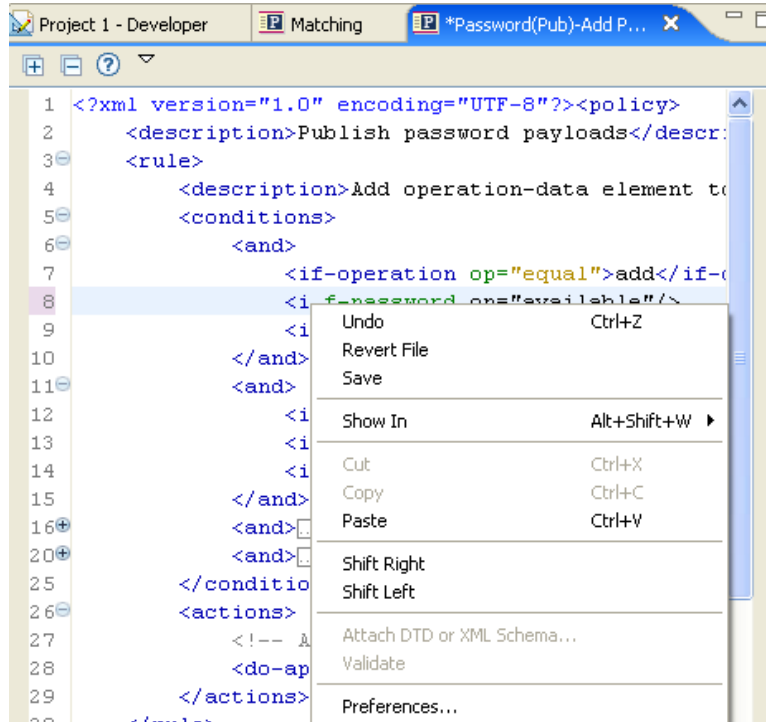


若要查看整個樹狀檢視，請展開列出的每個項目。

編輯 XML 來源

您可以透過 XML 編輯器編輯 XML。您也可以透過使用者圖形 (GUI) 介面在此處進行變更。

特性 2-12 編輯 XML 來源



此時會載入與 .xml 檔案類型關聯的預設編輯器。若找不到預設編輯器，則會載入系統的文字編輯器。您可以透過 XML 編輯器編輯 XML。

您也可以透過使用者圖形 (GUI) 介面在此處進行變更。

表格 2-5 XML 編輯器選項

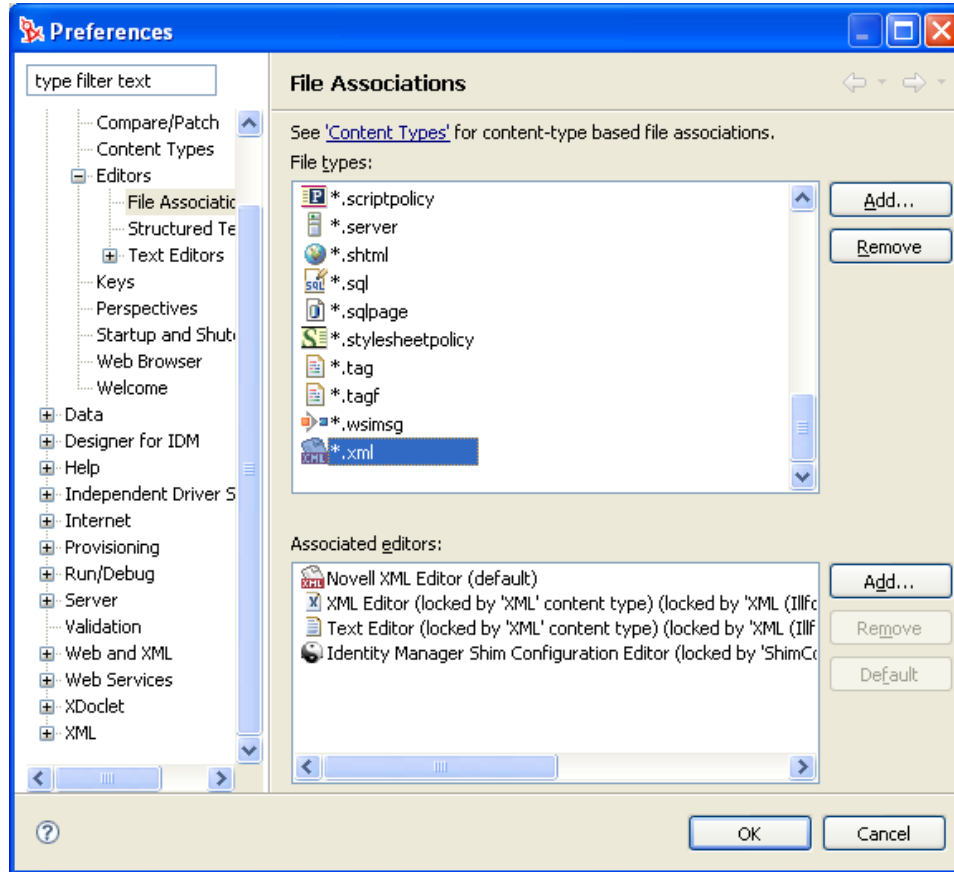
函數	描述
復原	復原上一個動作。
回復檔案	將檔案回復到上一次儲存的版本。
儲存	儲存檔案。
剪下	剪下選取的資訊。
貼上	將資訊貼至文件。
Shift Right	向右縮排。
Shift Left	向左縮排。
附加 DTD 或 XML 綱要	附加驗證規則的 DTD 或 XML 綱要檔案。
驗證	驗證 XML 程式碼。

優先設定

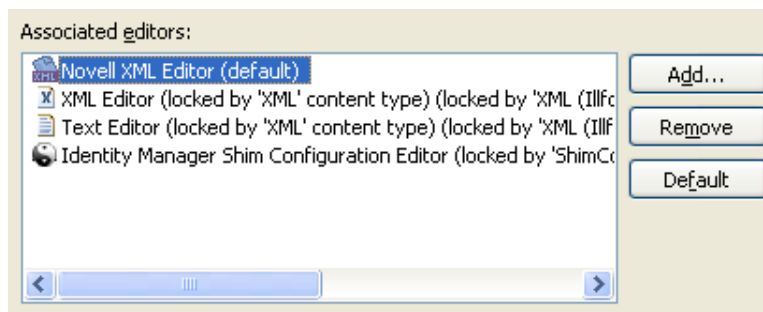
設定 XML 編輯器的優先設定。

若要為您的「來源」檢視窗選取不同的 XML 編輯器：

- 1 從「主要」功能表中選取「Window」>「優先設定」。
- 2 選取「一般」>「編輯器」>「檔案關聯」。
- 3 從「檔案類型」下的清單中選取「*.xml」。



- 4 在「關聯」編輯器窗格內選取所需的編輯器 (例如,「Novell XML Editor」)(若您要使用的編輯器未出現在清單中,您可以按一下「新增」,然後將其新增到清單中)。

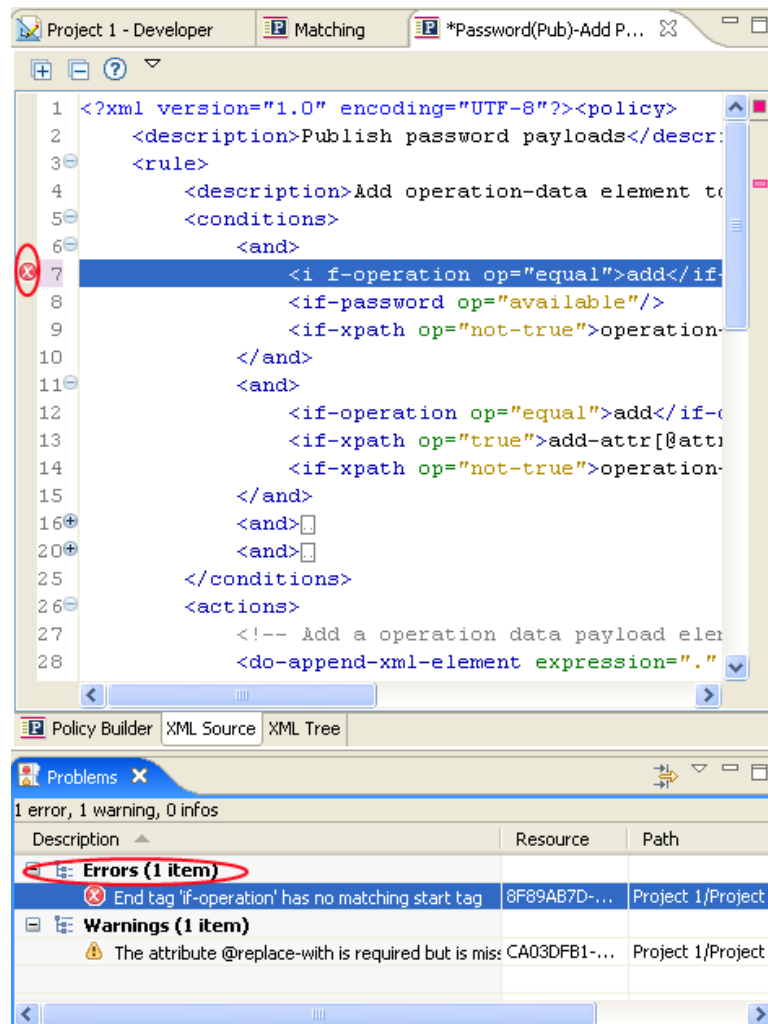


- 5 按一下「確定」。
- 6 關閉並重新開啓「規則產生器」。

驗證 XML 來源

XML 編輯器可用於驗證 XML 程式碼。按一下滑鼠右鍵，然後選取「驗證」。若出現錯誤，則在發生錯誤的該行上會顯示一個紅色的 x。視窗底端提供了與問題相關的詳細說明資訊。

特性 2-13 驗證 XML 來源



在此例中，其 if-operation 的結束標記與開始標記不符。

2.3 一般運算式

一般運算式是遵循某些模式之相符文字字串的公式。一般運算式是由一般字元和中繼字元所組成。一般字元包括大寫和小寫字母與數字。中繼字元具有特殊意義。下表包含最常見的中繼字元及其代表的意義。

表格 2-6 常見的一般運算式

中繼字元	描述
.	符合任何單一字元。
\$	符合一行的末端。
^	符合一行的起始處。
*	符合緊接著前方之零次或多次出現的字元。
\	逸出字元 此字元可讓您搜尋任何中繼字元。例如， \\$ 尋找 \$1000，而不是符合一行的末端。
[]	符合括號間的任一字元。
[0-9]	符合以連字號連接的字元範圍。此範例為符合任一數字。
[A-Za-z]	同時符合多個範圍。此範例為符合所有大寫及小寫的字母。

「引數產生器」專門設計為依照 Java 內定義的方式使用一般運算式。[Java 網站 \(http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html\)](http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html) 包含進一步的資訊。

2.4 XPath 1.0 運算式

某些條件、動作和記號使用 XPath 1.0 運算式的引數。XPath 是一種專為提供 XSLT 和 Xpointer 之間共用功能之常見語法及語意而建立的語言。其主要用於處理 XML 文件的部份，但也提供處理字串、數字和布林值的基本功能。

XPath 規格需要內嵌式應用程式提供網路位置與數個應用程式定義的訊息片段。在 DirXML 程序檔中 (請參閱「DirXML 程序檔」，第 11 頁)，XPath 用於評估以下的網路位置：

- ◆ 網路位置節點是目前的操作。
- ◆ 網路位置的位置及大小為 1。
- ◆ 其中內含數個可用的變數：
 - ◆ 可做為 Identity Manager 內樣式表參數 (目前有 fromNDS、srcQueryProcessor、destQueryProcessor、srcCommandProcessor、destCommandProcessor 和 dnConverter) 的變數。
 - ◆ 全域組態變數。
 - ◆ 本地規則變數。
 - ◆ 若不同的變數來源之間發生名稱上的衝突，則其優先順序為本地變數、樣式表參數、全域變數。
- ◆ 在規則元素上宣告名稱空間。
- ◆ 其中內含數個可用的功能：
 - ◆ 所有內建的 XPath 1.0 功能。
 - ◆ 由 NXSL 提供的 Java 延伸功能。

與具有 Java 類別字首關聯的名稱空間宣告必須在規則元素上宣告。

W3 網站 (<http://www.w3.org/TR/1999/REC-xpath-19991116>) 包含進一步的資訊。

2.5 條件

本節包含使用「規則產生器」介面時，可以使用到的所有條件。

- ◆ 「If 關聯」，第 114 頁
- ◆ 「If Attribute」，第 115 頁
- ◆ 「If Class Name」，第 116 頁
- ◆ 「If Destination Attribute」，第 117 頁
- ◆ 「If 目的 DN」，第 118 頁
- ◆ 「If Entitlement」，第 119 頁
- ◆ 「If Global Configuration Value」，第 120 頁
- ◆ 「If Local Variable」，第 121 頁
- ◆ 「If Named Password」，第 123 頁
- ◆ 「If Operation」，第 123 頁
- ◆ 「If Operation Attribute」，第 125 頁
- ◆ 「If Operation Property」，第 126 頁
- ◆ 「If Password」，第 127 頁
- ◆ 「If Source Attribute」，第 127 頁
- ◆ 「If Source DN」，第 128 頁
- ◆ 「If XPath Expression」，第 129 頁

2.5.1 If 關聯

對目前操作或目前物件的關聯值進行測試。

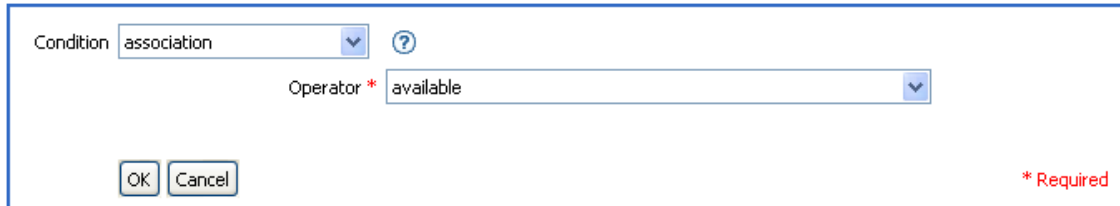
欄位

符合運算子條件的情況

運算子	符合條件的情況
associated	已為目前物件建立關聯。
available	目前操作指定了一個非空白關聯值。
equal	目前操作指定的關聯值完全等於 if 關聯的內容。
not-associated	尚未為目前物件建立關聯。
not available	關聯無法供目前物件使用。
not-equal	目前操作指定的關聯值不等於 if 關聯的內容。

範例

此例將測試關聯是否可用。當此條件符合時，即執行定義的動作。



Condition: association [?] [v]

Operator: * available [v]

OK Cancel

* Required

2.5.2 If Attribute

對目前操作或來源資料儲存內目前物件的屬性值進行測試。此屬性在邏輯上可被視為 If Operation Attribute 或 If Source Attribute，因為若符合來源資料儲存或操作內的條件，則測試結果是令人滿意的。

欄位

名稱

指定要測試的屬性名稱。

運算子

選取條件測試類型。

比較模式

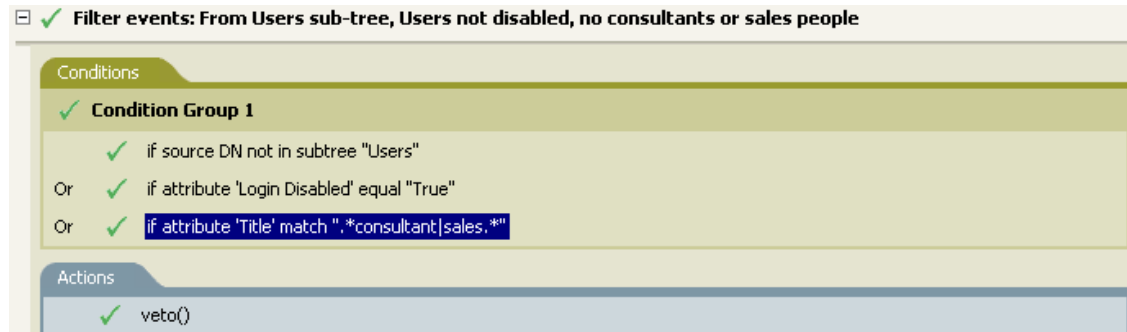
選取比較模式。請參閱「[比較模式](#)」，第 194 頁。

符合運算子條件的情況

運算子	符合條件的情況
available	指定屬性的目前操作或來源資料儲存中內含可用值。
equal	指定屬性的目前操作或來源資料儲存中內含可用值，當使用指定比較模式進行比較時，其等於指定值。
not available	available 將傳回 False。
not-equal	equal 將傳回 False。

範例

此範例在過濾已停用或具有某些職稱的「使用者」物件時，使用了 If Attribute 條件。該項規則為「過濾事件的規則」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。



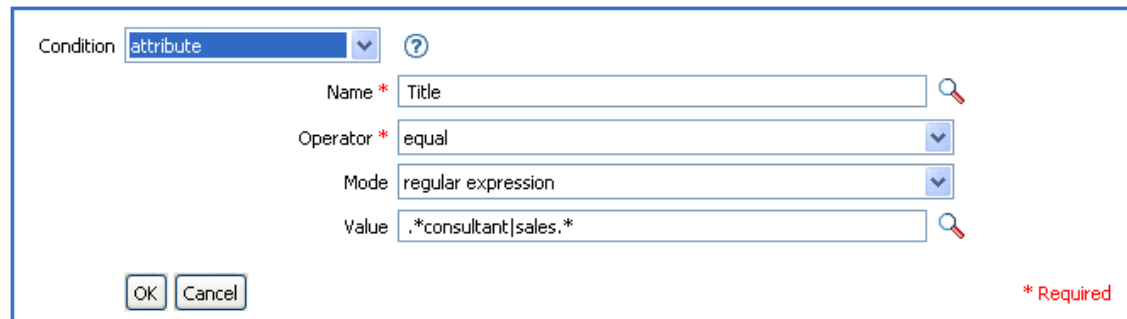
Filter events: From Users sub-tree, Users not disabled, no consultants or sales people

Conditions

- Condition Group 1
 - if source DN not in subtree "Users"
 - Or if attribute 'Login Disabled' equal "True"
 - Or if attribute 'Title' match ".*consultant|sales.*"

Actions

- veto()



Condition attribute

Name * Title

Operator * equal

Mode regular expression

Value .*consultant|sales.*

OK Cancel

* Required

此條件會尋找任何指定為顧問或銷售人員之具有 Title 屬性的「使用者」物件。

2.5.3 If Class Name

對目前操作的物件類別名稱進行測試。

欄位

運算子

選取條件測試類型。

比較模式

選取比較模式。請參閱「[比較模式](#)」，第 194 頁。

符合運算子條件的情況

運算子	符合條件的情況
available	目前操作中內含可用的物件類別名稱。
equal	目前操作中內含可用的物件類別名稱，並且在使用指定比較模式進行比較時，其等於指定值。
not available	available 將傳回 False。

運算子	符合條件的情況
not-equal	equal 將傳回 False。

範例

此範例使用 If Class Name 條件來為「使用者」物件依其職稱決定群組成員。該項規則為「依 Title 屬性管理使用者的群組」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

檢查目前操作的類別名稱是否為 User。

2.5.4 If Destination Attribute

對目前操作或目的資料儲存內目前物件的屬性值進行測試。

欄位

名稱

指定要測試的屬性名稱。

運算子

選取條件測試類型。

比較模式

選取比較模式。請參閱「[比較模式](#)」，第 194 頁。

符合運算子條件的情況

運算子	符合條件的情況
available	指定屬性的目的資料儲存中內含可用值。
equal	目的資料儲存內之指定屬性中內含可用值，當使用指定比較模式進行比較時，其等於指定值。
not available	available 將傳回 False。
not-equal	equal 將傳回 False。

範例

此範例使用 If Attribute 條件來為「使用者」物件依其職稱決定群組成員。該項規則為「依 Title 屬性管理使用者的群組」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

此規則會檢查 Title 屬性值是否包含 manager。

2.5.5 If 目的 DN

對目前操作的目的 DN 進行測試。

欄位

運算子

選取條件測試類型。

符合運算子條件的情況

運算子	符合條件的情況
available	其中內含可用的目的 DN。
equal	其中內含可用的目的 DN，並且當使用適用於目的資料儲存之 DN 格式的語意進行比較時，其等於指定值。
in-container	其中內含可用的目的 DN，並且當使用適用於目的資料儲存之 DN 格式的語意進行比較時，其代表容器內的某個物件，由值指定。
in-subtree	其中內含可用的目的 DN，並且當使用適用於目的資料儲存之 DN 格式的語意進行比較時，其代表子網路樹內的某個物件，由值指定。
not available	available 將傳回 False。
not-equal	equal 將傳回 False。
not-in-container	in-container 將傳回 False。
not-in-subtree	in-subtree 將傳回 False。

範例

Condition: destination DN

Operator *: in container

Value: Users

OK Cancel

* Required

2.5.6 If Entitlement

對目前操作或 Identity Vault 內目前物件的授權進行測試。

欄位

名稱

指定要為選取條件進行測試的授權名稱。

運算子

選取條件測試類型。

比較模式

選取比較模式。請參閱「[比較模式](#)」，第 194 頁。

符合運算子條件的情況

運算子	符合條件的情況
available	目前操作或 Identity Vault 內含可用的命名授權。
changing	目前操作包含命名授權的變更 (修改屬性或新增屬性)。

運算子	符合條件的情況
changing-from	目前操作包含命名授權的移除值變更 (移除值)，當使用指定比較模式進行比較時，其中內含的值等於指定值。
changing-to	目前操作包含對命名授權的新增值 (新增值或新增屬性) 變更。在使用指定比較模式進行比較時，其具有與指定值相等的值。
equal	目的資料儲存內之指定屬性中內含可用值，當使用指定比較模式進行比較時，其等於指定值。
not available	available 將傳回 False。
not-changing	changing 將傳回 False。
not-changing-from	changing-from 將傳回 False。
not-changing-to	changing-to 將傳回 False。
not-equal	equal 將傳回 False。

範例

Condition: entitlement

Name *: notes-group

Operator *: changing from

Mode: case insensitive

Value: Users

OK Cancel

* Required

2.5.7 If Global Configuration Value

對全域組態變數進行測試。

欄位

名稱

指定要為選取條件進行測試的全域變數名稱。

運算子

選取條件測試類型。

比較模式

選取比較模式。請參閱「[比較模式](#)」，第 194 頁。

符合運算子條件的情況

運算子	符合條件的情況
available	其中內含使用指定名稱的全域組態變數。

運算子	符合條件的情況
equal	其中內含使用指定名稱的全域組態變數，並且當使用指定比較模式進行比較時，其等於指定值。
not available	available 將傳回 False。
not-equal	equal 將傳回 False。

範例

2.5.8 If Local Variable

對本地變數進行測試。

欄位

名稱

指定要為選取條件進行測試的本地變數名稱。

運算子

選取條件測試類型。

比較模式

選取比較模式。請參閱「[比較模式](#)」，第 194 頁。

符合運算子條件的情況

運算子	符合條件的情況
available	其中內含本地變數，其指定名稱由規則 (Policy) 內舊版規則 (Rule) 的動作加以定義。
equal	其中內含使用指定名稱的本地變數，並且當使用指定比較模式進行比較時，其等於指定值。
not available	available 將傳回 False。
not-equal	equal 將傳回 False。

範例

此範例會依據 Title，將「使用者」物件新增至適當的群組 (Employee 或 Manager)。同時也會視需要建立群組，並設定與該群組相等的安全性。該項規則為「依 Title 屬性管理使用者」。

的群組」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

Set local variables to test existence of groups and for placement
 Create ManagersGroup, if needed

Conditions

Condition Group 1

- if local variable 'manager-group-info' available

And if local variable 'manager-group-info' not equal "group"

Actions

- add destination object(class name="Group", when="before", dn(Local Variable("manager-group-dn")))

Create EmployeesGroup, if needed
 If Title indicates Manager, add to ManagerGroup and set rights
 If Title does not indicate Manager, add to EmployeeGroup and set rights

此規則 (Policy) 包含五個互相倚賴的規則 (Rule)。

Set local variables to test existence of groups and for placement

Conditions

Condition Group 1

- if class name equal "User"

And

Condition Group 2

- if operation equal "add"

Or if operation equal "modify"

Actions

- set local variable("manager-group-dn", "Users\ManagersGroup")
- set local variable("manager-group-info", Destination Attribute("Object Class", dn(Local Variable("manager-group-dn"))))
- set local variable("employee-group-dn", "Users\EmployeesGroup")
- set local variable("employee-group-info", Destination Attribute("Object Class", dn(Local Variable("employee-group-dn"))))

爲了要讓 If Locate Variable 條件運作，第一個規則會設定四個用以測試群組的不同本地變數，以及放置群組的地方。

Condition: local variable ?

Name *

Operator * not equal

Mode case insensitive

Value

* Required

規則尋找的條件是要檢查 manager-group-info 的本地變數是否可用，以及 manager-group-info 是否不等於群組。若這些條件符合，則會新增群組的目的物件。

2.5.9 If Named Password

對使用指定名稱之目前操作的密碼進行測試。

欄位

名稱

指定要為選取條件進行測試的具名密碼名稱。

運算子

選取條件測試類型。

符合運算子條件的情況

運算子	符合條件的情況
available	其中內含使用可用之指定名稱的密碼。
not available	available 將傳回 False。

範例

Condition: named password (dropdown) [?]

Name *: password (text input) [magnifying glass]

Operator *: available (dropdown)

[OK] [Cancel] * Required

2.5.10 If Operation

對目前操作的名稱進行測試。

欄位

運算子

選取條件測試類型。

符合運算子條件的情況

運算子	符合條件的情況
equal	目前操作的名稱完全等於 If Operation 的內容。
not-equal	equal 將傳回 False。

值

其值為 Metadirectory 引擎在此條件中尋找的操作：

- ◆ add
- ◆ add-association
- ◆ check-object-password
- ◆ delete
- ◆ get-named-password
- ◆ modify
- ◆ modify-association
- ◆ modify-password
- ◆ move
- ◆ init-params
- ◆ instance

範例

此範例會依據 Title，將「使用者」物件新增至適當的群組 (Employee 或 Manager)。同時也會視需要建立群組，並設定與該群組相等的安全性。該項規則名稱為「依 Title 屬性管理使用者的群組」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

☑ **Set local variables to test existence of groups and for placement**

Conditions

- ✓ **Condition Group 1**
 - ✓ if class name equal "User"
- And**
- ✓ **Condition Group 2**
 - ✓ if operation equal "add"
 - Or ✓ if operation equal "modify"

Actions

- ✓ set local variable("manager-group-dn", "Users\ManagersGroup")
- ✓ set local variable("manager-group-info", Destination Attribute("Object Class", dn(Local Variable("manager-group-dn"))))
- ✓ set local variable("employee-group-dn", "Users\EmployeesGroup")
- ✓ set local variable("employee-group-info", Destination Attribute("Object Class", dn(Local Variable("employee-group-dn"))))

Condition ?

Operator *

Value 🔍

* Required

此條件會檢查是否已發生新增或修改操作。當發生這些事件之一時，會設定本地變數。

2.5.11 If Operation Attribute

對目前操作的屬性值進行測試。

欄位

名稱

指定要測試的屬性名稱。

運算子

選取條件測試類型。

比較模式

選取比較模式。請參閱「[比較模式](#)」，第 194 頁。

符合運算子條件的情況

運算子	符合條件的情況
available	指定屬性的目前操作 (新增屬性、新增值、屬性) 中內含可用值。
changing	目前操作包含指定屬性的變更 (修改屬性或新增屬性)。
changing-from	目前操作包含指定屬性的移除值 (移除值) 變更。在使用指定比較模式進行比較時，其等於指定值。
changing-to	目前操作包含對指定屬性的新增值 (新增值或新增屬性) 變更。在使用指定比較模式進行比較時，其等於指定值。
equal	指定屬性的目前操作 (除了移除值之外) 中內含可用值。在使用指定比較模式進行比較時，其等於指定值。
not available	available 將傳回 False。
not-changing	changing 將傳回 False。
not-changing-from	changing-from 將傳回 False。
not-changing-to	changing-to 將傳回 False。
not-equal	equal 將傳回 False。

範例

此範例會依據 Title，將「使用者」物件新增至適當的群組 (Employee 或 Manager)。同時也會視需要建立群組，並設定與該群組相等的安全性。該項規則名稱爲「依 Title 屬性管理使

用者的群組」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「可下載的 Identity Manager 規則」，第 32 頁。

The screenshot displays a rule configuration window with the following sections:

- Conditions:**
 - Condition Group 1
 - if class name equal "User"
 - And if operation attribute 'Title' match ".*manager.*"
- Actions:**
 - set destination attribute value("Group Membership", Local Variable("manager-group-dn"))
 - clone operation attribute("Group Membership", "Security Equals")

Below the actions, there is a checkbox for "If Title does not indicate Manager, add to EmployeeGroup and set rights".

The dialog box shows the configuration for a condition:

- Condition: destination attribute
- Name: Title
- Operator: equal
- Mode: regular expression
- Value: .*manager.*

Buttons: OK, Cancel. A red asterisk indicates that Name, Operator, and Value are required fields.

此條件會檢查 Title 屬性是否等於一般運算式 `.*manager*`。此條件會尋找在經理之前含零個或多個字元，以及經理之後含單一字元的職稱。若「使用者」物件的職稱為銷售經理，便會符合此條件。

2.5.12 If Operation Property

對目前操作的操作內容進行測試。

欄位

名稱

指定要為選取條件進行測試的操作內容名稱。

運算子

選取條件測試類型。

比較模式

選取比較模式。請參閱「比較模式」，第 194 頁。

符合運算子條件的情況

運算子	符合條件的情況
available	目前操作內含使用指定名稱的操作內容。
equal	目前操作內含使用指定名稱的操作內容，並且在使用指定比較模式進行比較時，其值等於所提供的內容。
not available	available 將傳回 False。
not-equal	equal 將傳回 False。

範例

2.5.13 If Password

對目前操作的密碼進行測試。

欄位

運算子

選取條件測試類型。

符合運算子條件的情況

運算子	符合條件的情況
available	目前操作中內含可用的密碼。
not available	available 將傳回 False。

範例

2.5.14 If Source Attribute

對目前操作或來源資料儲存內目前物件的屬性值進行測試。

欄位

名稱

指定要為選取條件進行測試的來源屬性名稱。

運算子

選取條件測試類型。

比較模式

選取比較模式。請參閱「[比較模式](#)」，第 194 頁。

符合運算子條件的情況

運算子	符合條件的情況
available	指定屬性的來源資料儲存中內含可用值。
equal	指定屬性的來源資料儲存中內含可用值。在使用指定比較模式進行比較時，其等於指定值。
not available	available 將傳回 False。
not-equal	equal 將傳回 False。

範例

Condition: source attribute

Name * : OU

Operator * : equal

Mode : case insensitive

Value : Users

OK Cancel

* Required

2.5.15 If Source DN

對目前操作的來源 DN 進行測試。

欄位

運算子

選取條件測試類型。

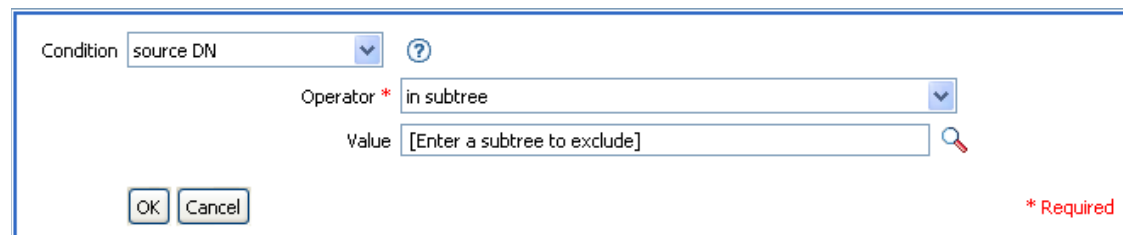
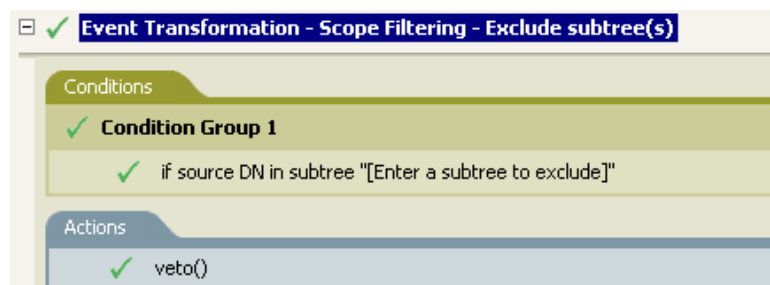
符合運算子條件的情況

運算子	符合條件的情況
available	DN 可用。

運算子	符合條件的情況
equal	其中內含可用的來源 DN，並且其等於指定值 in-container 的內容。其中內含可用的來源 DN，並且在指定值所識別的容器內，其代表一個物件。
in-subtree	其中內含可用的來源 DN，並且其代表由指定值所識別之子網路樹內的物件。
not available	available 將傳回 False。
not-equal	equal 將傳回 False。
not-in-container	in-container 將傳回 False。
not-in-subtree	in-subtree 將傳回 False。

範例

此範例使用 If Source DN 條件來檢查位於來源 DN 內的「使用者」物件。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[事件轉換：範圍過濾：排除子網路樹](#)」，第 80 頁。



此條件用於檢查 Users 容器內是否存在來源 DN。若物件來自該容器，則會被否決。

2.5.16 If XPath Expression

對 XPath 1.0 運算式的計算結果進行測試。

欄位


運算子

選取條件測試類型。



符合運算子條件的情況

運算子	符合條件的情況
true	XPath 運算式會評估為 True。
false	true 將傳回 False。

範例

Condition 

Operator *

Value  

2.6 動作

本節包含使用「規則產生器」介面時，可用的所有動作之詳細參考。

- ◆ 「新增關聯」，第 131 頁
- ◆ 「新增目的屬性值」，第 132 頁
- ◆ 「新增目的物件」，第 133 頁
- ◆ 「新增來源屬性值」，第 135 頁
- ◆ 「新增來源物件」，第 136 頁
- ◆ 「附加 XML 元素」，第 136 頁
- ◆ 「附加 XML 文字」，第 137 頁
- ◆ 「中斷」，第 138 頁
- ◆ 「清除目的屬性值」，第 138 頁
- ◆ 「清除操作內容」，第 139 頁
- ◆ 「清除來源屬性值」，第 139 頁
- ◆ 「清除單次登入 (SSO) 身分證明」，第 140 頁
- ◆ 「由 XPATH 運算式複製」，第 140 頁
- ◆ 「複製操作屬性」，第 141 頁
- ◆ 「刪除目的物件」，第 142 頁
- ◆ 「刪除來源物件」，第 143 頁
- ◆ 「尋找相符物件」，第 143 頁
- ◆ 「針對每個」，第 144 頁
- ◆ 「產生事件」，第 145 頁
- ◆ 「實作授權」，第 147 頁
- ◆ 「移動目的物件」，第 148 頁
- ◆ 「移動來源物件」，第 149 頁

- ◆ 「重新格式化操作屬性」，第 150 頁
- ◆ 「移除關聯」，第 151 頁
- ◆ 「移除目的屬性值」，第 152 頁
- ◆ 「移除來源屬性值」，第 152 頁
- ◆ 「重新命名目的物件」，第 153 頁
- ◆ 「重新命名操作屬性」，第 154 頁
- ◆ 「重新命名來源物件」，第 154 頁
- ◆ 「傳送電子郵件」，第 155 頁
- ◆ 「從範本傳送電子郵件」，第 156 頁
- ◆ 「設定預設的屬性值」，第 157 頁
- ◆ 「設定目的屬性值」，第 158 頁
- ◆ 「設定目的密碼」，第 159 頁
- ◆ 「設定本地變數」，第 160 頁
- ◆ 「設定操作關聯」，第 161 頁
- ◆ 「設定操作類別名稱」，第 162 頁
- ◆ 「設定操作目的 DN」，第 162 頁
- ◆ 「設定操作內容」，第 163 頁
- ◆ 「設定操作來源 DN」，第 164 頁
- ◆ 「設定操作範本 DN」，第 164 頁
- ◆ 「設定來源屬性值」，第 165 頁
- ◆ 「設定來源密碼」，第 166 頁
- ◆ 「設定 SSO 身分證明」，第 166 頁
- ◆ 「設定 SSO 通關密語」，第 167 頁
- ◆ 「設定 XML 屬性」，第 168 頁
- ◆ 「狀態」，第 168 頁
- ◆ 「條紋化操作屬性」，第 169 頁
- ◆ 「條紋化 XPath」，第 170 頁
- ◆ 「追蹤訊息」，第 170 頁
- ◆ 「否決」，第 171 頁
- ◆ 「若無法使用操作屬性則否決」，第 172 頁

2.6.1 新增關聯

以指定的關聯傳送新增關聯指令至 Identity Vault。

欄位

模式

選取此動作是否應被新增至目前操作，或直接寫入至 Identity Vault。

DN

指定目標物件的 DN 或保留為空白以使用目前物件。

關聯

指定要新增之關聯的值。

範例

The screenshot shows a configuration dialog box with the following elements:

- Do:** A dropdown menu with "add association" selected and a help icon (?) to its right.
- Select mode:** A dropdown menu with "add to current operation" selected.
- Information:** A dashed box containing an information icon (i) and the text "Leave the DN field below blank to use the current object".
- Enter DN:** A text input field containing "Source DN()" and a help icon (?) to its right.
- Enter association: *** A text input field containing "Source Name()" and a help icon (?) to its right.
- Buttons:** "OK" and "Cancel" buttons at the bottom left.

2.6.2 新增目的屬性值

在目的資料儲存內的物件上為屬性新增值。

欄位

屬性名稱

指定屬性的名稱。

類別名稱

(選擇性) 指定目標物件的類別名稱。保留為空白以使用目前物件的類別名稱。

模式

選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。

物件

選取目標物件。此物件可以是目前物件或由 DN 或關聯指定。

值類型

選取要新增之屬性值的語法。

值

指定要新增的屬性值。

範例

此範例會將目的屬性值新增到 OU 屬性。它會從已建立的本地變數建立其值。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[指令轉換：建立部門容器：第 1 部份和第 2 部份](#)」，第 69 頁。

Command Transformation - Create Departmental Container - Part 1

Conditions

- ✓ **Condition Group 1**
 - ✓ if operation equal "add"

Actions

- ✓ set local variable("target-container", Destination DN(length="-2"))
- set local variable("does-target-exist", Destination
- ✓ Attribute("objectclass", class name="Organizational Unit", dn(Local Variable("target-container"))))

Command Transformation - Create Departmental Container - Part 2

Conditions

- ✓ **Condition Group 1**
 - ✓ if local variable 'does-target-exist' available
 - And ✓ if local variable 'does-target-exist' equal ""

Actions

- ✓ add destination object(class name="organizational Unit", direct="true", dn(Local Variable("target-container")))
- add destination attribute value("ou", direct="true", dn(Local Variable("target-container")), Parse DN("dest-dn", "dot", length="1", start="-1", Local Variable("target-container")))
- ✓ add destination attribute value("ou", direct="true", dn(Local Variable("target-container")), Parse DN("dest-dn", "dot", length="1", start="-1", Local Variable("target-container")))

Do **add destination attribute value** ?

Enter attribute name: * 🔍

Enter class name: 🔍

Select mode: ▾

Select object: ▾

Enter DN: * 📄

Enter value type: ▾

Enter string: * 📄

* Required

2.6.3 新增目的物件

在目的資料儲存內建立新物件。

欄位

類別名稱

指定要建立之物件的類別名稱。

模式

選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。

DN

指定要建立之物件的 DN。

備註

任何要新增為建立物件之一部份的屬性值必須要在使用相同 DN 之後完成「[新增目的屬性值](#)」，[第 132 頁](#) 動作。

範例

此範例會建立所需的部門容器。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[指令轉換：建立部門容器：第 1 部份和第 2 部份](#)」，[第 69 頁](#)。

☐ ✓ **Command Transformation - Create Departmental Container - Part 1**

Conditions

- ✓ **Condition Group 1**
 - ✓ if operation equal "add"

Actions

- ✓ set local variable("target-container", Destination DN(length="-2"))
- set local variable("does-target-exist", Destination
- ✓ Attribute("objectclass", class name="Organizational Unit", dn(Local Variable("target-container"))))

☐ ✓ **Command Transformation - Create Departmental Container - Part 2**

Conditions

- ✓ **Condition Group 1**
 - ✓ if local variable 'does-target-exist' available
 - And ✓ if local variable 'does-target-exist' equal ""

Actions

- ✓ add destination object(class name="organizational Unit", direct="true", dn(Local Variable("target-container")))
- add destination attribute value("ou", direct="true", dn(Local Variable("target-container")), Parse DN("dest-dn", "dot", length="1", start="-1", Local Variable("target-container")))
- ✓

已建立「組織單位」物件。OU 屬性的值是從此動作後發生之目的屬性值動作所建立的。

2.6.4 新增來源屬性值

在來源資料儲存內的物件上為屬性新增值。

欄位

屬性名稱

指定屬性的名稱。

類別名稱

(選擇性) 指定目標物件的類別名稱。保留為空白以使用目前物件的類別名稱。

物件

選取目標物件。此物件可以是目前物件或由 DN 或關聯指定。

值類型

選取要新增之屬性值的語法。

值

指定要新增的屬性值。

範例

2.6.5 新增來源物件

建立要建立在來源資料儲存內的指定類型物件。任何要新增為建立物件之一部份的屬性值必須要在使用相同 DN 之後完成 [新增來源屬性值 \(第 135 頁\)](#) 動作。

欄位

類別名稱

指定要新增之物件的類別名稱。

DN

指定要新增之物件的 DN。

範例

The image shows two screenshots of a configuration interface. The top screenshot is for adding a source object. It has a title bar with a green checkmark and the text "add source object(class name='User', dn('Users\John Smith'))". Below the title bar, there is a "Do" dropdown menu with "add source object" selected. There are two input fields: "Enter class name: * User" and "Enter DN: * 'Users\John Smith'". There are "OK" and "Cancel" buttons at the bottom left, and a red asterisk with the word "Required" at the bottom right. The bottom screenshot is for adding a source attribute value. It has a title bar with a green checkmark and the text "add source attribute value('Title', class name='User', 'Manager')". Below the title bar, there is a "Do" dropdown menu with "add source attribute value" selected. There are four input fields: "Enter attribute name: * Title", "Enter class name: User", "Select object: Current object", and "Enter value type: string". There is one more input field: "Enter string: * 'Manager'". There are "OK" and "Cancel" buttons at the bottom left, and a red asterisk with the word "Required" at the bottom right.

2.6.6 附加 XML 元素

附加元素到 XPath 運算式選取的元素集中。

欄位

變數名稱

指定 XML 元素的標記名稱。若先前已在此規則中定義字首，則此名稱可以包含名稱空間字首。

XPath 運算式

指定傳回節點集的 XPath 1.0 運算式，此節點集包含應附加新元素的元素。

範例

✓ append XML element("jdbc:sql", "../jdbc:statement[last()])

Do ?

Enter variable name: * 🔍

Enter XPATH expression: * 🏠 ➡

* Required

✓ append XML text("../jdbc:statement[last()]/jdbc:sql", "UPDATE dixml.emp SET fname"+Operation Attribute("Member"))

Do ?

Enter XPATH expression: * 🏠 ➡

Enter string: * 📄

* Required

2.6.7 附加 XML 文字

附加文字到 XPath 運算式選取的元素集中。

欄位

XPath 運算式

XPath 1.0 運算式，此運算是可傳回包含應附加文字之元素的節點集。

字串

指定要附加的文字。

範例

The first screenshot shows a dialog box titled "append XML element('jdbc:sql', '..//jdbc:statement[last()'])". It has a "Do" dropdown set to "append XML element". Below it are two input fields: "Enter variable name: * jdbc:sql" and "Enter XPATH expression: * ..//jdbc:statement[last()]". There are "OK" and "Cancel" buttons at the bottom left, and a "* Required" label at the bottom right.

The second screenshot shows a dialog box titled "append XML text('..//jdbc:statement[last()]/jdbc:sql', 'UPDATE dixml.emp SET frame'+Operation Attribute('Member'))". It has a "Do" dropdown set to "append XML text". Below it are two input fields: "Enter XPATH expression: * ..//jdbc:statement[last()]/jdbc:sql" and "Enter string: * 'UPDATE dixml.emp SET frame'+Operation Attribute('Member)". There are "OK" and "Cancel" buttons at the bottom left, and a "* Required" label at the bottom right.

2.6.8 中斷

由目前規則結束目前操作的處理。

範例

The screenshot shows a dialog box with a "Do" dropdown set to "break". There are "OK" and "Cancel" buttons at the bottom left.

2.6.9 清除目的屬性值

移除來自目的資料儲存內物件的所有屬性值。

欄位

屬性名稱

指定屬性的名稱。

類別名稱

(選擇性) 指定目標物件的類別名稱。保留為空白以使用目前物件的類別名稱。


模式


選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。


物件


選取目標物件。此物件可以是目前物件或由 DN 或關聯指定。


範例

Do 

Enter attribute name: * 

Enter class name: 

Select mode: 

Select object: 

* Required

2.6.10 清除操作內容


清除目前操作的任何操作內容。

欄位

內容名稱

指定要清除之操作內容的名稱。

範例

Do 

Enter property name: *

2.6.11 清除來源屬性值

移除來自來源資料儲存內物件的所有屬性值。

欄位

屬性名稱

指定屬性的名稱。


類別名稱


(選擇性) 指定目標物件的類別名稱。保留為空白以使用目前物件的類別名稱。


物件


選取目標物件。此物件可以是目前物件或由 DN 或關聯指定。

範例

Do clear source attribute value 

Enter attribute name: * 

Enter class name: 

Select object: Current object 

* Required

2.6.12 清除單次登入 (SSO) 身分證明

清除「單次登入」身分證明，使物件可以被取消提供。此動作是「身分證明提供」規則的一部份。如需相關資訊，請參閱第 4 章「Novell 身分證明提供規則」，第 307 頁。

欄位

身分證明儲存物件 DN

指定儲存機制物件的 DN。

目標使用者 DN

指定目標使用者的 DN。


應用程式身分證明 ID


指定儲存在應用程式物件內的應用程式身分證明。

登入參數字串


指定應用程式的每一個登入參數。登入參數是儲存在應用程式物件內的驗證金鑰。

範例

Do clear SSO credential 


Enter credential store object DN: * 

Render browsed DN relative to policy

Enter target user DN: * 

[Populate the following from an application object](#)

Enter application credential ID: *

Enter login parameter strings: 

* Required

2.6.13 由 XPATH 運算式複製

附加由 XPath 運算式選取之 XML 節點集的深度副本至由另一個 XPath 運算式選取之元素集。

欄位


來源 XPath 運算式

指定傳回節點集 (包含要複製的節點) 的 XPath 1.0 運算式。



目的 XPath 運算式

指定傳回節點集 (包含要附加所複製節點的元素) 的 XPath 1.0 運算式。

範例

Do clone by XPATH expressions 

Enter source XPATH expression: * @*

Enter destination XPATH expression: * ../modify[last()]  

* Required

2.6.14 複製操作屬性

將目前操作內出現的所有屬性複製到目前操作內的不同屬性。

欄位

來源名稱

指定複製來源的屬性名稱。

目的名稱

指定複製目的的屬性名稱。

範例

此範例會依據 Title，將「使用者」物件新增至適當的群組 (Employee 或 Manager)。同時也會視需要建立群組，並設定與該群組相等的安全性。該項規則為「依 Title 屬性管理使用者」。

的群組」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「可下載的 Identity Manager 規則」，第 32 頁。

- Set local variables to test existence of groups and for placement**
- Create ManagersGroup, if needed**
- Create EmployeesGroup, if needed**
- If Title indicates Manager, add to ManagerGroup and set rights**

Conditions

- Condition Group 1**
- if class name equal "User"
- And if operation attribute 'Title' match ".*manager.*"

Actions

- set destination attribute value("Group Membership", Local Variable("manager-group-dn"))
- clone operation attribute("Group Membership", "Security Equals")

- If Title does not indicate Manager, add to EmployeeGroup and set rights**

Do clone operation attribute ?

Enter source name: * 🔍

Enter destination name: 🔍

* Required

「複製操作屬性」會取得來自 Group Membership 屬性的資訊，並將該資訊新增至 Security Equals 屬性，使其值相同。

2.6.15 刪除目的物件

在目的資料儲存內刪除物件。

欄位

模式

選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。

物件

選取目標物件。此物件可以是目前物件或由 DN 或關聯指定。

範例

The screenshot shows a dialog box with a blue border. At the top left, there is a dropdown menu labeled 'Do' with the text 'delete destination object' and a question mark icon to its right. Below this, there are two more dropdown menus: 'Select mode:' with the text 'add to current operation' and 'Select object:' with the text 'Current object'. At the bottom left, there are two buttons: 'OK' and 'Cancel'. At the bottom right, there is a red asterisk followed by the text '* Required'.

2.6.16 刪除來源物件

在來源資料儲存內刪除物件。

欄位

模式

選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。

物件

在來源資料儲存內選取要刪除的目標物件。此物件可以是目前物件或由 DN 或關聯指定。

範例

The screenshot shows a dialog box with a blue border. At the top left, there is a dropdown menu labeled 'Do' with the text 'delete source object' and a question mark icon to its right. Below this, there are two input fields: 'Select object:' with the text 'DN' and a dropdown arrow, and 'Enter DN: *' with the text '"Uses\John Smith"' and a list icon. At the bottom left, there are two buttons: 'OK' and 'Cancel'. At the bottom right, there is a red asterisk followed by the text '* Required'.

2.6.17 尋找相符物件

在目的資料儲存內尋找與目前物件相符的物件。

欄位

範圍

選取搜尋的範圍。搜尋範圍可以是一個項目、從屬物件或子網路樹。

DN

指定搜尋的基本 DN。

相符屬性

指定要搜尋的屬性值。

備註

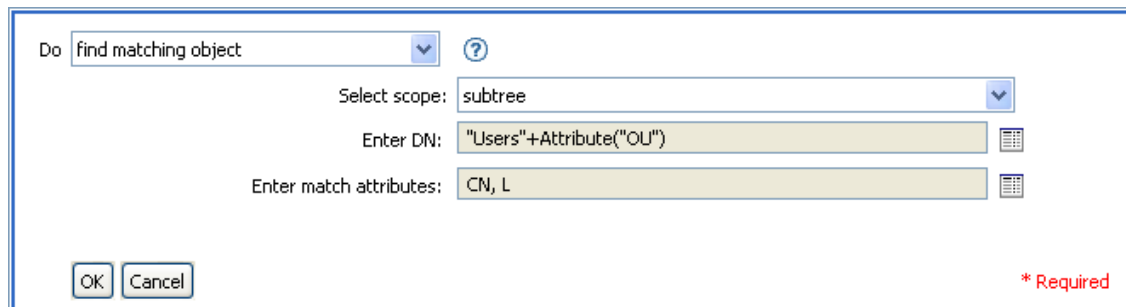
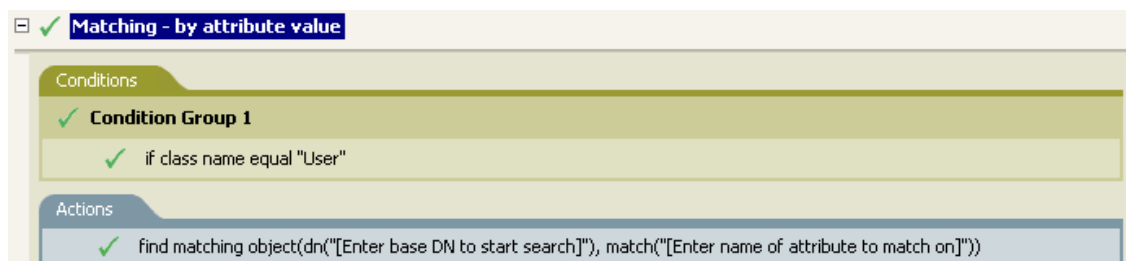
「尋找相符物件」僅在目前操作為新增操作時有效。

當範圍為「項目」時需要使用 DN 引數，其他範圍則可選擇性使用。當範圍為「子網路樹」或「從屬物件」時，至少需要使用一個相符屬性。若範圍為「項目」，並且已指定相符屬性，則結果會不明確。如果目的資料儲存為已連接應用程式，則會在目前操作中為所傳回的每個成功相符項目新增關聯。若目前操作已擁有非空白關聯，則不會執行查詢，因此可允許多個尋找相符物件動作串接在相同規則中。

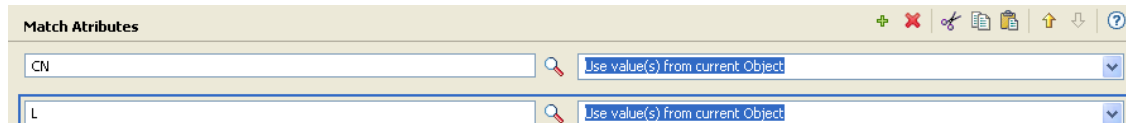
若目的資料儲存為 Identity Vault，則會設定目前操作的目的 DN 屬性。若目前操作已擁有非空白的目的 DN 屬性，則不會執行查詢，因此可允許多個尋找相符物件動作串接在相同規則中。若僅傳回單一結果，並且其尚未關聯，則目前操作的目的 DN 會設定為相符物件的來源 DN。若僅傳回單一結果，並且其已關聯，則目前操作的目的 DN 會設定為單一字元 �。若傳回多重結果，則目前操作的目的 DN 會設定為單一字元 。

範例

此範例會符合使用屬性 CN 和 L 的「使用者」物件。搜尋規則的位置會從 Users 容器開始，並且將儲存在 OU 屬性內的資訊新增到 DN。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「比對：依屬性值」，第 87 頁。



當您按一下「引數產生器」圖示時，便會啟動「相符屬性產生器」。您可以在產生器中指定想要比對的屬性。在此例中使用了 CN 和 L 屬性。



2.6.18 針對每個

為節點集內的每個節點重複一組動作。

欄位

節點集

指定節點集。

動作

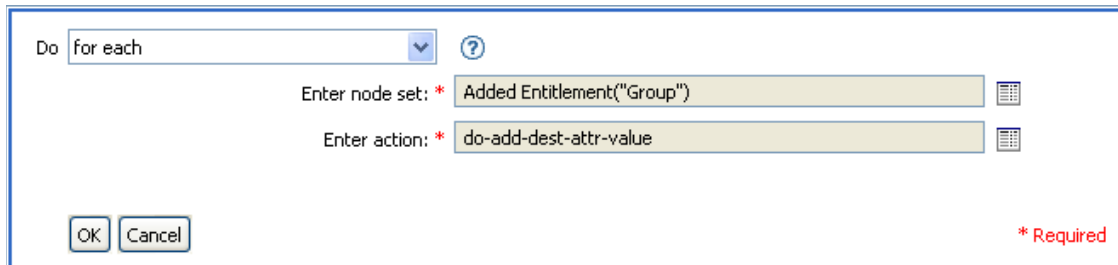
指定要在節點集內的每個節點上執行的動作。

備註

若使用了本地變數，則目前節點的值與每個重複動作的值不同。

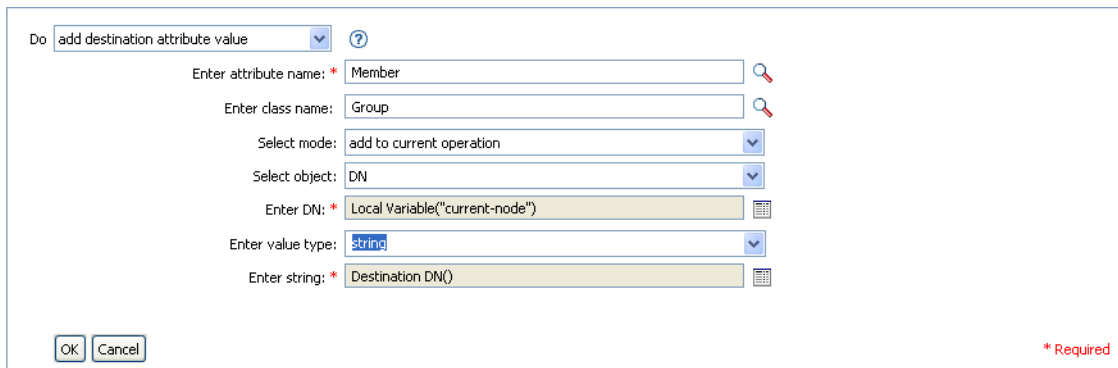
如果節點集內的某個節點為授權，則「針對每個」會以隱含方式執行「實作授權」，第 147 頁動作。

範例



The screenshot shows a dialog box with a 'Do' dropdown menu set to 'for each'. Below it, there are two input fields: 'Enter node set: *' with the value 'Added Entitlement("Group")' and 'Enter action: *' with the value 'do-add-dest-attr-value'. At the bottom left are 'OK' and 'Cancel' buttons, and at the bottom right is a red asterisk followed by the text '* Required'.

以下為用來提供動作引數的「引數動作產生器」範例：



The screenshot shows a dialog box titled 'add destination attribute value'. It contains several fields: 'Enter attribute name: *' with 'Member', 'Enter class name:' with 'Group', 'Select mode:' with 'add to current operation', 'Select object:' with 'DN', 'Enter DN: *' with 'Local Variable("current-node")', 'Enter value type:' with 'string', and 'Enter string: *' with 'Destination DN()'. There are search icons next to the class name and DN fields. At the bottom left are 'OK' and 'Cancel' buttons, and at the bottom right is a red asterisk followed by the text '* Required'.

2.6.19 產生事件

傳送使用者定義的事件至 Novell Audit。

欄位

ID

指定事件的 ID。ID 必須為 1000-1999 範圍內的整數。

層級

選取事件的層級。

層級	描述
log-emergency	導致 Metadirectory 引擎或驅動程式關閉的事件。
log-alert	需要立即注意的事件。
log-critical	可導致部份 Metadirectory 引擎或驅動程式故障的事件。
log-error	描述可由 Metadirectory 引擎或驅動程式處理之錯誤的事件。
log-warning	不代表問題的負面事件。
log-notice	管理員可用來瞭解或改善使用和操作的事件 (正面或負面事件)。
log-info	任何重要的正面事件。
log-debug	可讓支援或工程師除錯 Metadirectory 引擎或驅動程式操作的相關事件。

字串

指定要包含在事件內之使用者定義的字串、整數和二進位值。這些值會由「具名字串產生器」提供。

字串名稱	描述
target	將對其採取動作的物件。
target-type	為目標指定預先定義格式的整數。target-type 目前預先定義的值為： <ul style="list-style-type: none"> ◆ 0 = 無 ◆ 1 = 斜線標記 ◆ 2 = 點標記 ◆ 2 = LDAP 標記
subTarget	將對其採取動作之目標的子元件。
text1	在此處輸入的文字會被儲存在 text1 事件欄位中。
text2	在此處輸入的文字會被儲存在 text2 事件欄位中。
text3	在此處輸入的文字會被儲存在 text3 事件欄位中。
value	在此處輸入的任何數字都會被儲存在 value 事件欄位中。
value3	在此處輸入的任何數字都會被儲存在 value3 事件欄位中。
data	在此處輸入的資料會被儲存在 blob 事件欄位中。

備註

Novell Audit 事件結構包含一個 target、一個 subTarget、三個字串 (text1、text2、text3)、兩個整數 (value、value3)，以及一個一般欄位 (data)。文字欄位限制為 256 個位元組，同時 data 欄位最多可以包含 3 KB 的資訊，除非在您的環境中啟用了較大的 data 欄位。

範例

此範例有四個規則 (Rule)，這些規則根據 Surname 屬性的第一個字元實作「使用者」物件的「佈置」規則 (Policy)。它會產生追蹤訊息和自定的 Novell Audit 事件。「產生事件」動

作用於傳送 Novell Audit 事件。該項規則的名稱為「依姓放置的規則」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「可下載的 Identity Manager 規則」，第 32 頁。

☑ **Setup Local Variables**

☑ **Surname A-I: place in Users1**

Conditions

- ☑ **Condition Group 1**
- ☑ if class name equal "User"
- And ☑ if operation attribute 'Surname' match "[a-].*"

Actions

- ☑ set operation destination DN(dn("Training\Users\Active\Users1"+"\"+Operation Attribute("CN")))
- ☑ trace message(color="yellow", Local Variable("LVUsers1"))
- ☑ generate event(id="1000", text1=Local Variable("LVUsers1"))

☑ **Surname J-R: place in Users2**

☑ **Surname S-Z: place in Users3**

Do generate event

Enter ID: * 1000

Select level: informational

Enter strings: text1

OK Cancel

* Required

以下為使用「具名字串產生器」提供字串引數的範例。

Name	String Value
text1	Local Variable("LVUsers1")

「產生事件」會建立一個名為 ID 1000 的事件，並顯示由 LVUser1 的本地變數所產生的文字。本地變數 LVUser1 是一個新增至 "+"Training\Users\Active\Users1"+" 容器的 User:Operation Attribute "cn" +" 字串。此事件會讀取新增至 Training\Users\Active\Users1 容器的 User:jsmith 事件。

2.6.20 實作授權

指定實作授權的動作，使那些授權的狀態可以被回報至已授予或已撤銷授權的代理程式。

欄位

節點集

包含由指定動作實作之授權的節點集。

動作

實作指定授權的動作。

範例

The dialog box has a title bar and a main area with the following fields:

- Do:** A dropdown menu with the value "implement entitlement" and a help icon.
- Enter node set: *** A text input field containing "Removed Entitlement("Account")" and a list icon.
- Enter action: *** A text input field containing "do-add-dest-attr-value" and a list icon.

At the bottom left are "OK" and "Cancel" buttons. At the bottom right is a red asterisk followed by the text "* Required".

以下為用來提供動作引數的「引數動作產生器」範例：

The dialog box has a title bar and a main area with the following fields:

- Do:** A dropdown menu with the value "add destination attribute value" and a help icon.
- Enter attribute name: *** A text input field containing "Login Disabled" and a search icon.
- Enter class name:** A text input field containing "User" and a search icon.
- Select mode:** A dropdown menu with the value "add to current operation".
- Select object:** A dropdown menu with the value "DN".
- Enter DN: *** A text input field containing "Local Variable("current-node")" and a list icon.
- Enter value type:** A dropdown menu with the value "string".
- Enter string: *** A text input field containing "Destination DN()" and a list icon.

At the bottom left are "OK" and "Cancel" buttons. At the bottom right is a red asterisk followed by the text "* Required".

2.6.21 移動目的物件

在目的資料儲存內移動物件。

欄位

模式

選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。

類別名稱

(選擇性) 指定要移動之物件的類別名稱。保留為空白以使用目前物件的類別名稱。

要移動的物件

選取要被移動的物件。此物件可以是目前物件或者可以由 DN 或關聯指定。

要移至的容器

選取目標容器。此容器可以由 DN 或關聯指定。

範例

此範例包含單一規則，此規則會在 **Description** 屬性指出使用者的帳戶為已終止時，停用該帳戶並移至已停用容器。該項規則被命名為「在終止時停用並移動使用者帳戶」，可從

Novell 的支援網站下載。如需相關資訊，請參閱「可下載的 Identity Manager 規則」，第 32 頁。

The screenshot shows the configuration for a rule titled "On Termination, disable user and move to Disabled container". It is divided into two sections: "Conditions" and "Actions".

Conditions:

- Condition Group 1
 - if operation equal "modify"
 - And if class name equal "User"
 - And if operation attribute 'Description' match "^terminated.*"

Actions:

- set destination attribute value("Login Disabled", direct="true", "True")
- move destination object(when="after", dn("Users\Disabled"))

The screenshot shows a dialog box for configuring the "move destination object" action. It includes the following fields:

- Do:** move destination object
- Select mode:** add after current operation
- Select object to move:** Current object
- Select container to move to:** DN
- Enter DN: *** "Users\Disabled"

Buttons for "OK" and "Cancel" are at the bottom left. A red asterisk and the text "* Required" are at the bottom right.

此規則會檢查「使用者」物件是否發生修改事件，以及 Description 屬性是否包含已終止的值。若情況如此，則將 Login Disabled 的屬性設定為 true，並將物件移至 User\Disabled 容器。

2.6.22 移動來源物件

在來源資料儲存內移動物件。

欄位

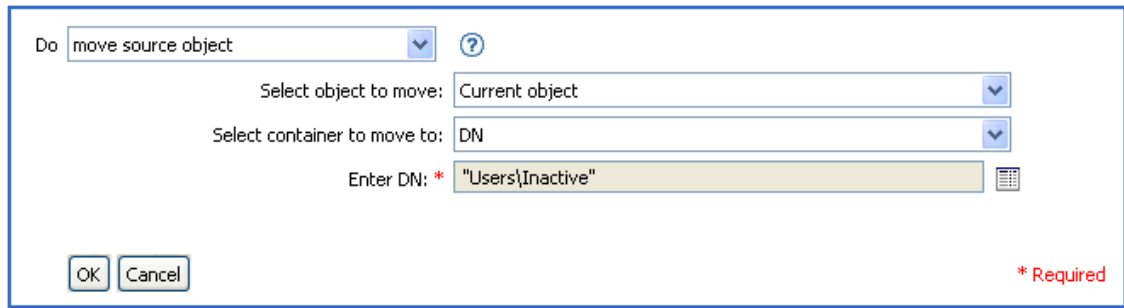
要移動的物件

選取要被移動的物件。此物件可以是目前物件或者可以由 DN 或關聯指定。

要移至的容器

選取目標容器。此容器可以由 DN 或關聯指定。

範例



2.6.23 重新格式化操作屬性

重新格式化使用某個模式之目前操作內的所有屬性。

欄位

名稱

指定屬性的名稱。

值類型

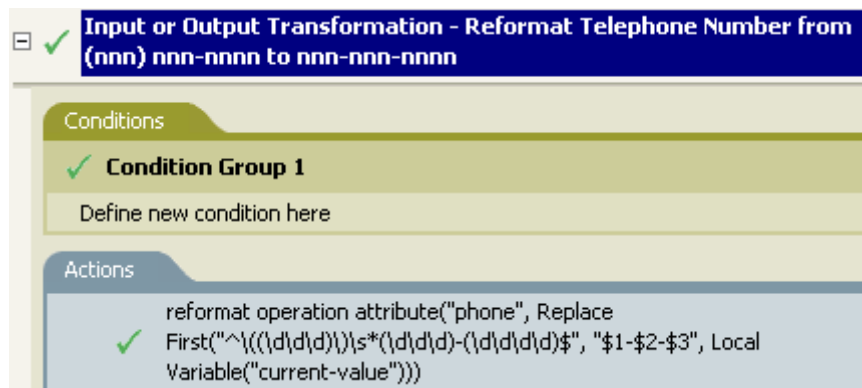
指定新屬性值的語法。

值

指定要用為屬性值之新格式的模式值。若原始值需要建構新值，則必須藉由參考變數 `current-value` 來取得該值。

範例

此範例會重新格式化電話號碼。電話號碼會由 `(nnn)-nnn-nnnn` 變更為 `nnn-nnn-nnnn`。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「輸入或輸出轉換：將電話號碼從 `(nnn) nnn-nnnn` 重新格式化為 `nnn-nnn-nnnn`」，第 81 頁。



重新格式化操作屬性動作會變更電話號碼的格式。此規則會使用「引數產生器」及一般運算式來變更顯示資訊的方式。

2.6.24 移除關聯

傳送移除關聯指令至 Identity Vault。

欄位

模式

選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。

關聯

指定要移除之關聯的值。

範例

此範例會採取刪除操作並以停用「使用者」物件做為取代。並進行事件的轉換。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[指令轉換：要停用的發行者刪除](#)」，第 71 頁。

在發生「使用者」物件的刪除操作時，「禁止登入」的屬性值會被設定為 `true`，並且會移除與物件的關聯。由於已連接應用程式內的關聯的物件已不存在，因此會移除關聯。

2.6.25 移除目的屬性值

從目的資料儲存內的物件上移除屬性值。

欄位

屬性名稱

指定屬性的名稱。

類別名稱

(選擇性) 指定目標物件的類別名稱。保留為空白以使用目前物件的類別名稱。

模式

選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。

物件

選取目標物件 此物件可以是目前物件或者可以由 DN 或關聯指定。

值類型

指定要移除之屬性值的語法。

值

指定新屬性值。

範例

The screenshot shows a configuration dialog box with the following fields and options:

- Do:** A dropdown menu set to "remove destination attribute value".
- Enter attribute name: *** A text input field containing "Title".
- Enter class name:** A text input field containing "User".
- Select mode:** A dropdown menu set to "add to current operation".
- Select object:** A dropdown menu set to "Current object".
- Enter value type:** A dropdown menu set to "string".
- Enter string: *** A text input field containing "Destination DN()".

At the bottom left are "OK" and "Cancel" buttons. At the bottom right is a red asterisk followed by the text "* Required".

2.6.26 移除來源屬性值

從來源資料儲存中物件上的具名屬性移除指定的值。

欄位

屬性名稱

指定屬性的名稱。

類別名稱

(選擇性) 指定目標物件的類別名稱。保留為空白以使用目前物件的類別名稱。

物件

選取目標物件 此物件可以是目前物件或者可以由 DN 或關聯指定。

值類型

指定要移除之屬性值的語法

值

指定要移除的屬性值。

範例

The screenshot shows a dialog box with the following fields and controls:

- Do:** A dropdown menu with the text "remove source attribute value" and a question mark icon.
- Enter attribute name: *** A text input field containing "Title" and a search icon.
- Enter class name:** A text input field containing "User" and a search icon.
- Select object:** A dropdown menu with "Current object" selected.
- Enter value type:** A dropdown menu with "string" selected.
- Enter string: *** A text input field containing "Destination DN()" and a list icon.
- Buttons:** "OK" and "Cancel" buttons at the bottom left.
- Footer:** "* Required" text at the bottom right.

2.6.27 重新命名目的物件

將目的資料儲存中的物件重新命名

欄位

模式

選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。

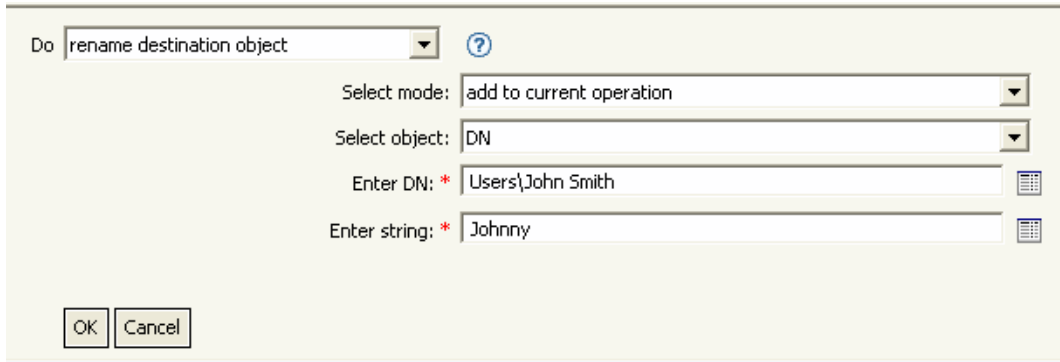
物件

選取目標物件 此物件可以是目前物件或者可以由 DN 或關聯指定。

字串

指定物件的新名稱。

範例



2.6.28 重新命名操作屬性

將目前操作中出現的所有屬性重新命名。

欄位

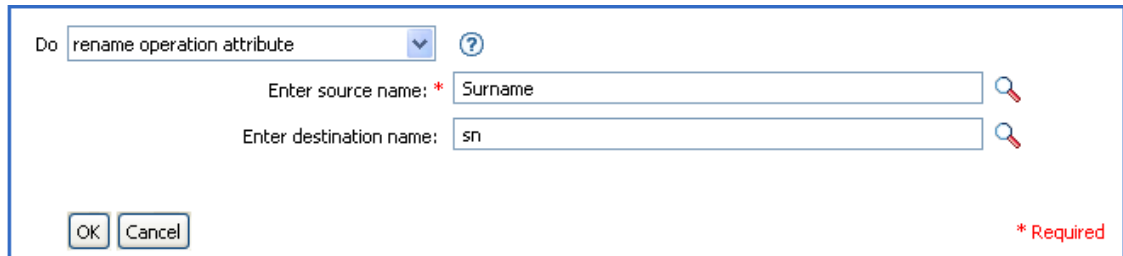
來源名稱

指定原始的屬性名稱。

目的名稱

指定新的屬性名稱。

範例



2.6.29 重新命名來源物件

重新命名來源資料儲存內的物件。

欄位

物件

選取目標物件 此物件可以是目前物件或由 DN 或關聯指定。

字串

指定物件的新名稱。

範例

Do: rename source object

Select object: DN

Enter DN: * "Users\John Smith"

Enter string: * "Johnny"

OK Cancel

* Required

2.6.30 傳送電子郵件

傳送電子郵件通知。

欄位

ID

(選擇性) 指定以 SMTP 系統傳送訊息的「使用者 ID」。

伺服器

指定 SMTP 伺服器名稱。

密碼

(選擇性) 指定 SMTP 伺服器的帳戶密碼。

重要：密碼屬性的值會以純文字儲存。

類型

選取電子郵件訊息類型。

字串

指定包含各種電子郵件地址、主旨和訊息的值。下表列出有效的具名字串引數：

字串名稱	描述
to	新增地址到電子郵件收件者的清單中；允許使用多個例項。
cc	新增地址到 CC (副本) 電子郵件收件者的清單中；允許使用多個例項。
bcc	新增地址到 BCC (密件副本) 電子郵件收件者的清單中；允許使用多個例項。
from	指定要做為寄出電子郵件地址的地址。
reply-to	指定要做為電子郵件訊息回覆地址的地址。
subject	指定電子郵件主旨。
message	指定電子郵件訊息的內容。
encoding	指定電子郵件訊息使用的字元編碼。

範例

Do: send email

Enter ID: user

Enter server: * smtp.company.com

Enter password: [masked]

Select message type: text

Enter strings: to, cc, bcc, from, subject, message

OK Cancel

* Required

以下為使用「具名字串產生器」提供字串引數的範例：

Name	String Value
to	"to_user1@company.com"
cc	"cc_user@company.com"
bcc	"bcc_user@company.com"
from	"from_user@company.com"
subject	""This is the e-mail subject""
message	"This is the e-mail body"

2.6.31 從範本傳送電子郵件

使用範本產生電子郵件通知。

欄位

通知 DN

以斜線形式指定 SMTP 通知組態物件的 DN。

範本 DN

以斜線形式指定電子郵件範本物件的 DN。

密碼

(選擇性) 指定 SMTP 伺服器的帳戶密碼。

重要：密碼屬性的值會以純文字儲存。

字串

指定電子郵件訊息的其他欄位。下列表格包含指定各種電子郵件地址的保留欄位名稱：

字串名稱	描述
to	新增地址到電子郵件收件者的清單中；允許使用多個例項。

字串名稱	描述
cc	新增地址到 CC (副本) 電子郵件收件者的清單中；允許使用多個例項。
bcc	新增地址到 BCC (密件副本) 電子郵件收件者的清單中；允許使用多個例項。
reply-to	指定要做為電子郵件訊息回覆地址的地址。
encoding	指定電子郵件訊息使用的字元編碼。

各個範本也會定義電子郵件訊息的主旨和本文中，可以被取代的欄位。

範例

以下為使用「具名字串產生器」提供字串引數的範例：

2.6.32 設定預設的屬性值

如果屬性尚未有值，則對目前操作（以及選擇性地對來源資料儲存中的目前物件）新增預設值。此動作僅在目前操作為新增操作時有效。

欄位

屬性名稱

指定預設屬性的名稱。

寫回

選取是否也要將預設值寫回來源資料儲存。

值

指定屬性的預設值。

範例

此範例會為屬性「公司」設定預設值。您可以自行選擇要設定的屬性值。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[建立：設定預設屬性值](#)」，第 75 頁。

Creation - Set Default Attribute Value

Conditions

- Condition Group 1
 - if class name equal "User"

Actions

- set default attribute value("[Enter attribute name]", write-back="true", "[Enter default attribute value"])

Do: set default attribute value

Enter attribute name: * company

Write back: true

Enter argument values: * "Digital Airlines Inc."

OK Cancel * Required

Type	Argument Values
string	"Digital Airlines Inc."

為了建構值，「引數值清單產生器」會被啟動。如需產生器的相關資訊，請參閱「[引數值清單產生器](#)」，第 61 頁。您可以視需要設定值。此例中使用了「引數產生器」，並且將文字設定為公司的名稱。

2.6.33 設定目的屬性值

為目的資料儲存內物件上的屬性新增值，並且移除該屬性的所有其他值。

欄位

屬性名稱

指定屬性的名稱。

類別名稱

(選擇性) 指定目的資料儲存內目標物件的類別名稱。保留為空白以使用目前物件的類別名稱。

模式

選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。

物件

選取目標物件 此物件可以是目前物件或由 DN 或關聯指定。

值類型

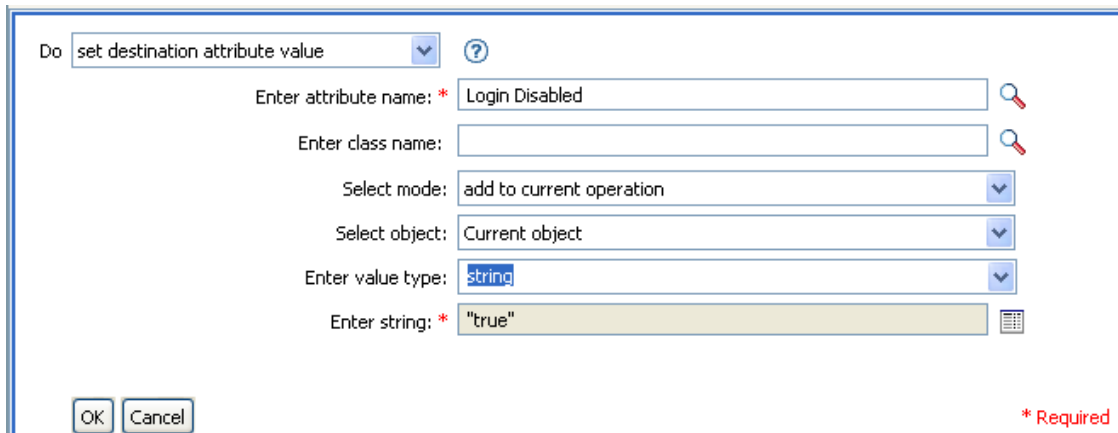
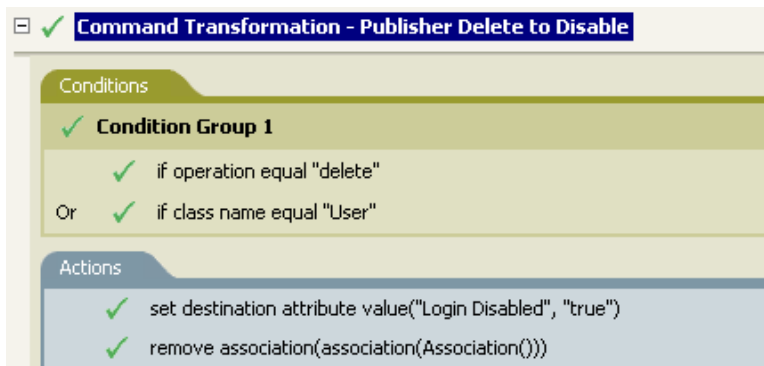
選取要設定之屬性值的語法。

值

指定要設定的屬性值。

範例

此範例會採取刪除操作並以停用「使用者」物件做為取代。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[指令轉換：要停用的發行者刪除](#)」，第 71 頁。



此規則會將「禁止登入」的屬性值設定為 true。此規則會使用「引數產生器」為屬性值新增 true 文字。如需與產生器相關的相關資訊，請參閱「[引數產生器](#)」，第 58 頁。

2.6.34 設定目的密碼

為目的資料儲存中的目前物件設定密碼。

欄位

模式

選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。

物件

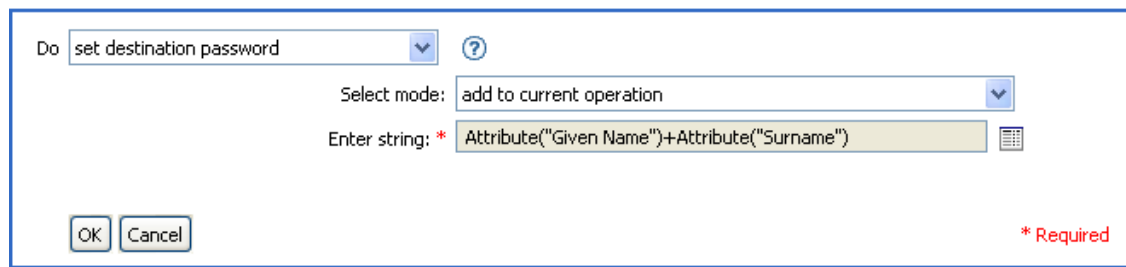
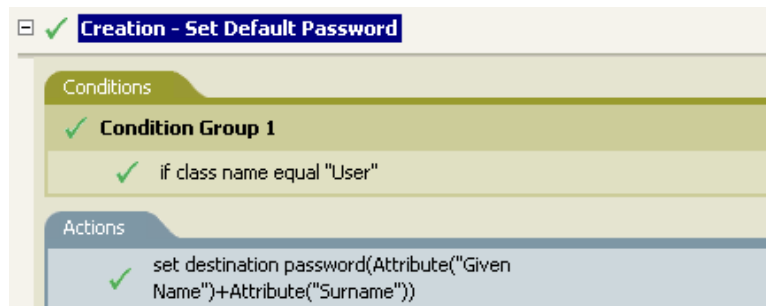
選取目標物件 此物件可以是目前物件或由 DN 或關聯指定。

字串

指定要設定的密碼。

範例

此範例會為建立的「使用者」物件設定預設的密碼。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[建立：設定預設密碼](#)」，第 77 頁。



當「使用者」物件被建立時，其密碼會被設定為 Given Name 屬性加上 Surname 屬性的組合。

2.6.35 設定本地變數

設定本地變數。

欄位

變數名稱

指定本地變數的名稱。

變數類型

選取本地變數的類型。此類型可以為字串、XPath 1.0 節點集或 Java 物件。

值

指定本地變數的值。

範例

此範例會依據 Title，將「使用者」物件新增至適當的群組 (Employee 或 Manager)。同時也會視需要建立群組，並設定與該群組相等的安全性。此規則名稱為「依 Title 管理使用者的

群組」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

The screenshot displays a rule configuration window titled "Set local variables to test existence of groups and for placement". It is divided into two main sections: "Conditions" and "Actions".

Conditions:

- Condition Group 1:** if class name equal "User"
- And**
- Condition Group 2:** if operation equal "add"
- Or** if operation equal "modify"

Actions:

- set local variable("manager-group-dn", "Users\ManagersGroup")
- set local variable("manager-group-info", Destination Attribute("Object Class", dn(Local Variable("manager-group-dn"))))
- set local variable("employee-group-dn", "Users\EmployeesGroup")
- set local variable("employee-group-info", Destination Attribute("Object Class", dn(Local Variable("employee-group-dn"))))

The screenshot shows the "set local variable" dialog box. It contains the following fields and controls:

- Do:** set local variable (dropdown menu)
- Enter variable name:** * LVUsers1 (text input field)
- Select variable type:** String (dropdown menu)
- Enter string:** * "User:"+Operation Attribute("cn")+ added to the "+"Training\ (text input field)
- Buttons:** OK, Cancel
- Footer:** * Required

本地變數會設定為「使用者」物件之「物件類別」加上 manager-group-info 的「本地變數」值。「引數產生器」會被用於建構本地變數。如需相關資訊，請參閱「[引數產生器](#)」，第 58 頁。

2.6.36 設定操作關聯

為目前操作設定關聯值。

欄位

關聯

提供新的關聯值。

範例

A dialog box with a blue border. At the top left, there is a dropdown menu labeled 'Do' with the text 'set operation association' and a question mark icon to its right. Below this, the text 'Enter association: *' is followed by a text input field containing 'Source Name()' and a list icon. At the bottom left are 'OK' and 'Cancel' buttons. At the bottom right, the text '* Required' is displayed in red.

2.6.37 設定操作類別名稱

為目前操作設定物件類別名稱。

欄位

字串

提供新的類別名稱。

範例

A dialog box with a blue border. At the top left, there is a dropdown menu labeled 'Do' with the text 'set operation class name' and a question mark icon to its right. Below this, the text 'Enter string: *' is followed by a text input field containing '"User"' and a list icon. At the bottom left are 'OK' and 'Cancel' buttons. At the bottom right, the text '* Required' is displayed in red.

2.6.38 設定操作目的 DN

為目前操作設定目的 DN。

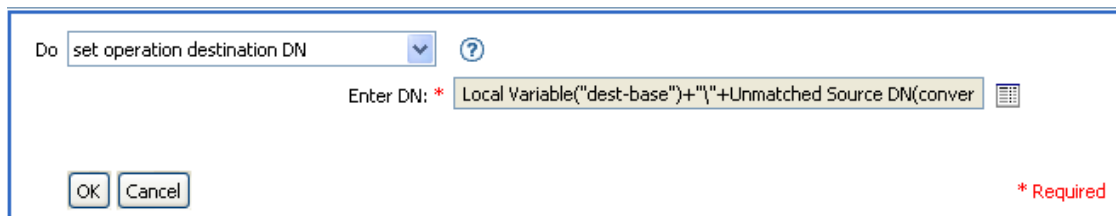
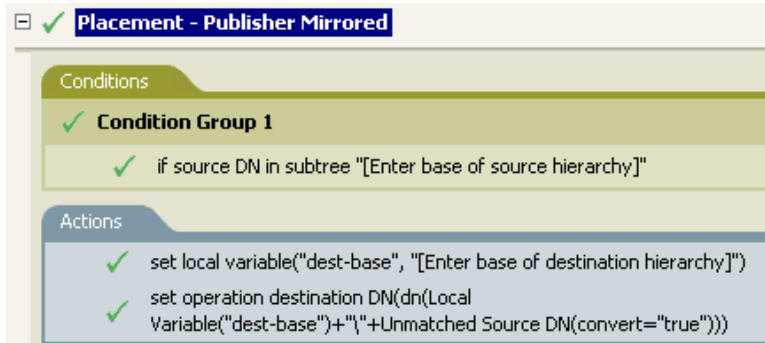
欄位

DN

指定新的目的 DN。

範例

此範例會使用來自已連接系統之鏡像複製的結構，將物件放置在 Identity Vault 內。您必須定義鏡像複製在來源和目的資料儲存中的哪一點開始。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[建立：設定預設屬性值](#)」，第 75 頁。



此規則會將操作目的 DN 設定目的基礎位置加上來源 DN 的本地變數。

2.6.39 設定操作內容

設定操作內容。操作內容是已儲存在某項操作內的具名值。一般用於提供處理操作結果之規則所需的其他網路位置。

欄位

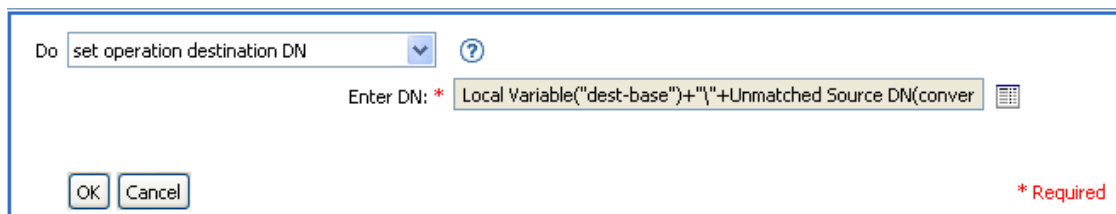
設定操作內容

指定操作內容的名稱。

字串

指定操作內容的名稱。

範例



2.6.40 設定操作來源 DN

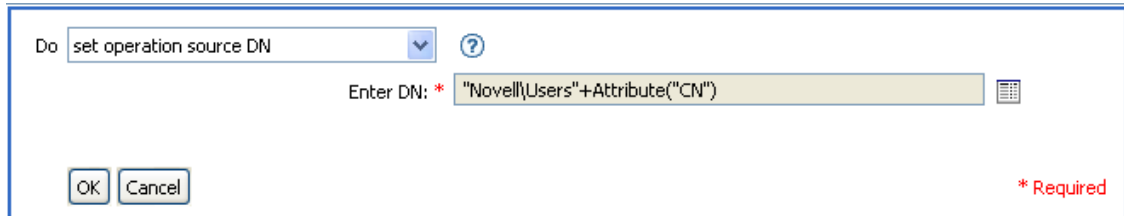
為目前操作設定來源 DN。

欄位

DN

指定新的來源 DN。

範例



Do: set operation source DN [?] [v]

Enter DN: * "Novell\Users"+Attribute("CN") [📄]

OK Cancel * Required

2.6.41 設定操作範本 DN

為目前操作設定範本 DN 為指定值。此動作僅在目前操作為新增操作時有效。

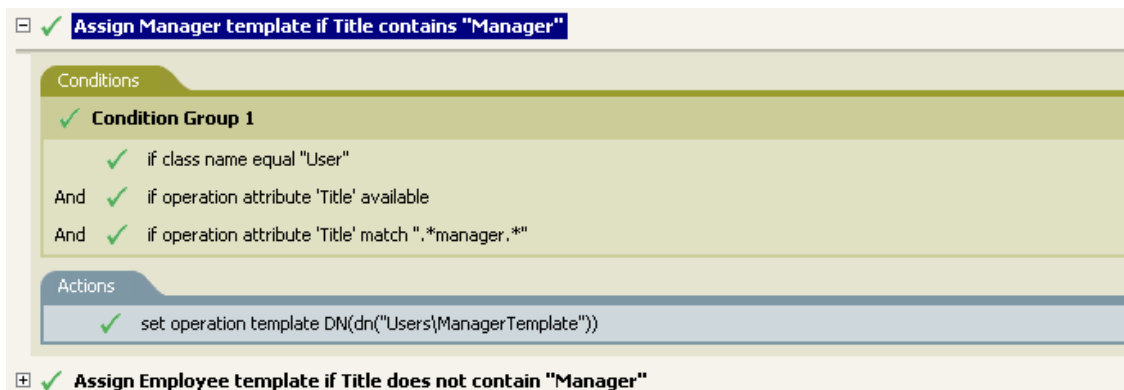
欄位

DN

指定範本 DN。

範例

若 Title 屬性包含 Manager 一字，此範例會套用 Manager 範本。該規則的名稱為「規則：依 Title 指定使用者的範本」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。



Assign Manager template if Title contains "Manager"

Conditions

Condition Group 1

- if class name equal "User"
- And if operation attribute 'Title' available
- And if operation attribute 'Title' match ".*manager.*"

Actions

- set operation template DN(dn("Users\ManagerTemplate"))

Assign Employee template if Title does not contain "Manager"

Do set operation template DN

Enter DN: * "Users\ManagerTemplate"

OK Cancel

* Required

範本 Manager Template 會被套用至任何有 Title 屬性可用，且 Title 中含有 manager 一字的「使用者」物件。此規則會使用一般運算式尋找所有可能相符的項目。

2.6.42 設定來源屬性值

為來源資料儲存內物件上的屬性新增值，並且移除該屬性的所有其他值。

欄位

屬性名稱

指定屬性的名稱。

類別名稱

(選擇性) 指定來源資料儲存內目標物件的類別名稱。保留為空白以使用目前物件的類別名稱。

物件

選取目標物件。此物件可以是目前物件或由 DN 或關聯指定。

值類型

選取屬性值的語法。

值

指定要設定的屬性值。

範例

此範例會偵測電子郵件地址的變更，並且將其設回其原始值。該規則的名稱為「規則：重設電子郵件屬性值」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

Push back on email changing

Conditions

Condition Group 1

- if class name equal "User"
- And if operation attribute "Email" changing

Actions

- set source attribute value("Email", Destination Attribute("Internet EMail Address"))
- strip operation attribute("Email")

The dialog box is titled 'Do' and contains the following fields:

- Do:** A dropdown menu with 'set source attribute value' selected.
- Enter attribute name: *** A text input field containing 'Email'.
- Enter class name:** An empty text input field.
- Select object:** A dropdown menu with 'Current object' selected.
- Enter value type:** A dropdown menu with 'string' selected.
- Enter string: *** A text input field containing 'Destination Attribute("Internet EMail Address")'.

At the bottom left are 'OK' and 'Cancel' buttons. At the bottom right is a red asterisk followed by the text '* Required'.

此動作會取目的屬性 Internet EMail Address 的值，並將「電子郵件」的來源屬性設定為這個值。

2.6.43 設定來源密碼

在來源資料儲存內為目前物件設定密碼。

欄位

字串

指定要設定的密碼。

範例

The dialog box is titled 'Do' and contains the following fields:

- Do:** A dropdown menu with 'set source password' selected.
- Enter string: *** A text input field containing 'Attribute("Given Name")+Attribute("Surname")'.

At the bottom left are 'OK' and 'Cancel' buttons. At the bottom right is a red asterisk followed by the text '* Required'.

2.6.44 設定 SSO 身分證明

在建立使用者物件或修改密碼時，設定 SSO 身分證明。此動作是「身分證明提供」規則的一部份。如需相關資訊，請參閱第 4 章「Novell 身分證明提供規則」，第 307 頁。

欄位

身分證明儲存物件 **DN**

指定儲存機制物件的 DN。

目標使用者 **DN**

指定目標使用者的 DN。

應用程式身分證明 **ID**

指定儲存在應用程式物件內的應用程式身分證明。

登入參數字串

指定應用程式的每一個登入參數。登入參數是儲存在應用程式物件內的驗證金鑰。

範例

Do **set SSO credential** ?

Enter credential store object DN: * 🔍
 Render browsed DN relative to policy

Enter target user DN: * 📄
[Populate the following from an application object](#)

Enter application credential ID: *

Enter login parameter strings: 📄

* Required

2.6.45 設定 SSO 通關密語

設定「Novell 安全登入®」通關密語，並在提供「使用者」時提供回答。此動作是「身分證明提供」規則的一部份。如需相關資訊，請參閱第 4 章「Novell 身分證明提供規則」，第 307 頁。

欄位

身分證明儲存物件 DN

指定儲存機制物件的 DN。

目標使用者 DN

指定目標使用者的 DN。

問題與回答字串

指定「安全登入」通關密語的問題和回答。

範例


Do **set SSO passphrase** ?

Enter credential store object DN: * 🔍
 Render browsed DN relative to policy

Enter target user DN: * 📄

Enter question and answer strings: 📄

* Required

「安全登入」通關密語的問題和回答會以字串型式儲存在規則中。按一下「編輯字串」圖示 ，以啟動字串產生器。指定通關密語的問題和回答。

2.6.46 設定 XML 屬性

在 XPath 運算式選取的元素集上設定 XML 屬性。

欄位

名稱

指定 XML 屬性的名稱。若先前已在此規則中定義字首，則此名稱可以包含名稱空間字首。

XPath 運算式

傳回節點集 (包含其上應設定 XML 屬性的元素) 的 XPath 1.0 運算式。

字串

指定 XML 屬性的值。

範例

The image displays two screenshots of a configuration dialog box for setting XML attributes. Each dialog has a title bar with a green checkmark and the XPath expression: `set XML attribute("cert-id", ".", "c:\lotus\domino\data\eng.id")` for the first and `set XML attribute("cert-pwd", ".", "certify2eng")` for the second. The dialog contains a dropdown menu set to "set XML attribute", a "Do" button, and three input fields: "Enter variable name: * cert-id", "Enter XPATH expression: * .", and "Enter string: * \"c:\lotus\domino\data\eng.id\"". There are also "OK" and "Cancel" buttons and a "* Required" label in the bottom right corner.

2.6.47 狀態

產生狀態通知。

欄位

層級

指定通知的狀態層級。

訊息

使用「引數產生器」提供的狀態訊息。

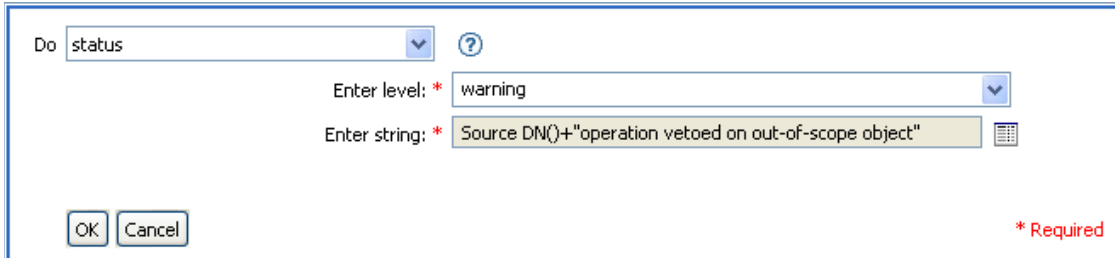
備註

若層級為 `retry`，則規則會立即終止輸入文件的處理程序，並為目前正在處理的事件的排定重試程序。

若層級為 `fatal`，則規則會立即停止輸入文件的處理程序，並啓始驅動程式的關閉程序。

若目前操作有一個 `event-id`，則會使用該 `event-id` 進行狀態通知，否則不會回報 `event-id`。

範例



Do: status

Enter level: * warning

Enter string: * Source DN()+ "operation vetoed on out-of-scope object"

OK Cancel

* Required

2.6.48 條紋化操作屬性

條紋化來自目前操作中所有出現的屬性。

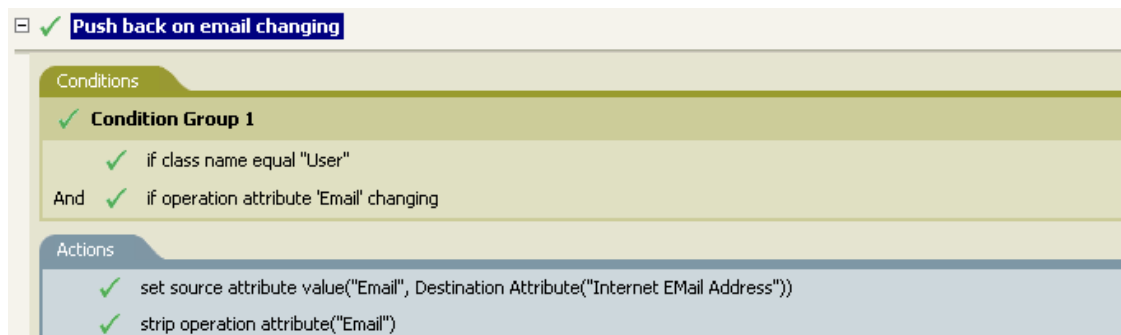
欄位

名稱

指定要條紋化的屬性名稱。

範例

此範例會偵測電子郵件地址的變更，並且將其設回其原始值。該規則的名稱為「規則：重設電子郵件屬性值」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。



Push back on email changing

Conditions

Condition Group 1

- if class name equal "User"
- And if operation attribute "Email" changing

Actions

- set source attribute value("Email", Destination Attribute("Internet EMail Address"))
- strip operation attribute("Email")

此動作會條紋化電子郵件的屬性。其中保留的值是目的電子郵件屬性內保留的值。

2.6.49 條紋化 XPath

條紋化由 XPath 運算式選取的節點。

欄位

XPath 運算式

指定傳回節點集 (包含要去除的節點) 的 XPath 1.0 運算式。

範例

2.6.50 追蹤訊息

傳送訊息到 DSTRACE。

欄位

層級

指定訊息的追蹤層級。預設的層級為 0。只有在指定的追蹤層級小於或等於驅動程式內組態的追蹤層級時，才會出現訊息。

如需如何在驅動程式上設定追蹤層級的相關資訊，請參閱《*Novell Identity Manager 3.0 管理指南*》內的「[檢視 Identity Manager 程序](#)」。

色彩

選取追蹤訊息的色彩。

字串

指定追蹤訊息的數值。

範例

此範例有四個規則 (Rule)，這些規則根據 Surname 屬性的第一個字元實作「使用者」物件的「佈置」規則 (Policy)。可以產生追蹤訊息和自定的 Novell Audit 事件。「追蹤訊息」動

作用於傳送追蹤訊息到 DSTRACE 中。該項規則的名稱為「依姓放置的規則」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「可下載的 Identity Manager 規則」，第 32 頁。

The screenshot shows a rule configuration window with the following sections:

- Setup Local Variables** (expanded)
- Surname A-I: place in Users1** (selected)
- Conditions**
 - Condition Group 1
 - if class name equal "User"
 - And if operation attribute 'Surname' match "[a-].*"
- Actions**
 - set operation destination DN(dn("Training\Users\Active\Users1"+"\"+Operation Attribute("CN")))
 - trace message(color="yellow", Local Variable("LVUsers1"))
 - generate event(id="1000", text1=Local Variable("LVUsers1"))
- Surname J-R: place in Users2
- Surname S-Z: place in Users3

The dialog shows the configuration for the 'Do' action:

- Do: trace message
- Enter level: []
- Select color: yellow
- Enter string: * Local Variable("LVUsers1")
- Buttons: OK, Cancel
- Footer: * Required

此動作會傳送追蹤訊息到 DSTRACE。本地變數的內容為 LVUsers1，並且在 DSTRACE 中會以黃色顯示。

2.6.51 否決

否決目前的操作。

範例

此範例會排除來自指定子網路樹的所有事件。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「事件轉換：範圍過濾：排除子網路樹」，第 80 頁。

The screenshot shows a rule configuration window with the following sections:

- Event Transformation - Scope Filtering - Exclude subtree(s)** (selected)
- Conditions**
 - Condition Group 1
 - if source DN in subtree "[Enter a subtree to exclude]"
- Actions**
 - veto()

Do veto [v] (?)

OK Cancel

* Required

此動作會否決來自指定子網路樹的所有事件。

2.6.52 若無法使用操作屬性則否決

依目前操作內屬性的可用性，有條件地取消目前的操作及結束目前規則的處理程序。

欄位

名稱

指定屬性的名稱。

範例

此範例不允許建立所有的「使用者」物件，除非可以使用 Given Name、Surname、Title、Description 和 Internet EMail Address 屬性。規則名稱為「強制屬性存在的規則」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「可下載的 Identity Manager 規則」，第 32 頁。

☐ ✓ **User required attributes: First/Last Name, Title, Description, Email**

Conditions

✓ **Condition Group 1**

✓ if class name equal "User"

Actions

- ✓ veto if operation attribute not available("Given Name")
- ✓ veto if operation attribute not available("Surname")
- ✓ veto if operation attribute not available("Title")
- ✓ veto if operation attribute not available("Description")
- ✓ veto if operation attribute not available("Internet EMail Address")

Do veto if operation attribute not available [v] (?)

Enter name: * Given Name [magnifying glass]

OK Cancel

* Required

如果無法使用 Given Name、Surname、Title、Description 和 Internet Email Address 屬性，此動作會否決該項操作。

2.7 名詞記號

本節包含使用「引數產生器」介面時，可用的所有名詞記號之詳細參考資訊。

- ◆ 「新增的授權」，第 173 頁
- ◆ 「關聯」，第 174 頁
- ◆ 「屬性」，第 174 頁
- ◆ 「類別名稱」，第 175 頁
- ◆ 「目的屬性」，第 175 頁
- ◆ 「目的 DN」，第 176 頁
- ◆ 「目的名稱」，第 177 頁
- ◆ 「授權」，第 177 頁
- ◆ 「全域組態值」，第 178 頁
- ◆ 「本地變數」，第 178 頁
- ◆ 「具名密碼」，第 179 頁
- ◆ 「操作」，第 179 頁
- ◆ 「操作屬性」，第 180 頁
- ◆ 「操作內容」，第 180 頁
- ◆ 「密碼」，第 181 頁
- ◆ 「移除的屬性」，第 181 頁
- ◆ 「移除的授權」，第 181 頁
- ◆ 「來源屬性」，第 181 頁
- ◆ 「來源 DN」，第 182 頁
- ◆ 「來源名稱」，第 182 頁
- ◆ 「文字」，第 183 頁
- ◆ 「唯一名稱」，第 184 頁
- ◆ 「Unmatched Source DN」，第 185 頁
- ◆ 「XPath」，第 186 頁

2.7.1 新增的授權

展開至目前操作中被授予的授權值。

欄位

名稱

授權的名稱。

範例

.....  Added Entitlement("manager")

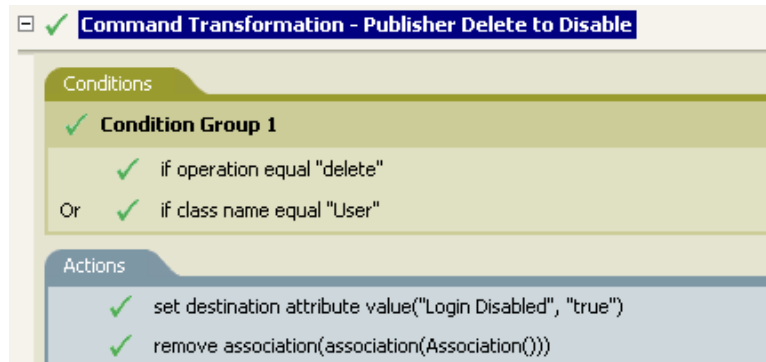
2.7.2 關聯

展開至目前操作中的關聯值。

範例

此範例取自 Identity Manager 3.0 隨附的預先定義規則。如需預先定義規則的相關資訊，請參閱「[指令轉換：要停用的發行者刪除](#)」，第 71 頁。

「移除關聯」的動作會使用「關聯」記號，從目前操作擷取其值。此規則會從「使用者」物件移除關聯，因此任何到達的新事件不會影響到「使用者」物件。



Association()

2.7.3 屬性

展開至目前操作和來源資料儲存中，來自目前物件的屬性值。此屬性在邏輯上可被視為操作屬性記號和來源屬性記號的結合 (union)。其中不包含從修改操作中移除的值。

欄位

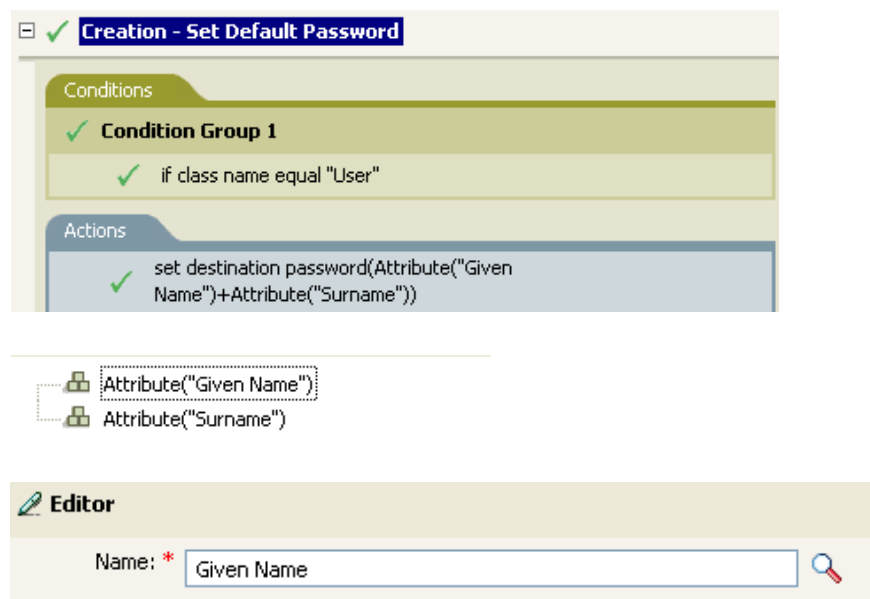
名稱

指定屬性的名稱。

範例

此範例取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[建立：設定預設密碼](#)」，第 77 頁。

「設定目的密碼」的動作使用屬性記號來建立密碼。密碼由 Given Name 屬性和 Surname 屬性所組成。當您在使用「引數產生器編輯器」時，可以瀏覽並選取想要使用的屬性。



2.7.4 類別名稱

展開至目前操作的物件類別名稱。

範例

..... `Class Name()`

2.7.5 目的屬性

展開至目的資料儲存內，目前操作、DN 或關聯的指定屬性值。

欄位

類別名稱

(選擇性) 指定目標物件的類別名稱。保留為空白以使用目前物件的類別名稱。

名稱

屬性名稱。

範例

此範例來自於「依 Title 管理使用者的群組」規則，可從 Novell 的支援網站下載。如需相關資訊，請參閱「可下載的 Identity Manager 規則」，第 32 頁。

該規則會以「引數產生器」建立「目的屬性」。「設定本地變數」的動作包含 Destination Attribute 記號。

此圖顯示了規則配置界面的一部分。在「Conditions」區域，有一個名為「Condition Group 1」的條件組，包含兩個條件：「if local variable 'manager-group-info' available」和「if local variable 'manager-group-info' not equal "group"」。在「Actions」區域，有一個動作：「add destination object(class name="Group", when="before", dn(Local Variable("manager-group-dn")))」。

Destination Attribute("Object Class", dn())

此圖顯示了「Destination Attribute」的編輯器。它包含以下欄位：

- 名稱：* 物件類別
- 類別名稱：
- 選取物件：DN
- 輸入DN：* Local Variable("manager-group-dn")

您可以透過「編輯器」建立 Destination Attribute。在此範例中，設定了「物件類別」的屬性。DN 用於選取目標物件。DN 的值是 manager-group-dn 的「本地變數」。

2.7.6 目的 DN

展開至目前操作中的目的 DN。

欄位

轉換

選擇是否要將 DN 轉換為來源資料儲存所使用的格式。

開始

指定要開始處理的相對可辨識名稱 (Relative Distinguished Name, RDN) 索引：

- ◆ 索引 0 為最根部的 RDN
- ◆ 正索引為最根部之 RDN 的偏移
- ◆ 索引 -1 為最葉部的段落
- ◆ 負索引為從最葉部之 RDN 朝向最根部之 RDN 的偏移

長度

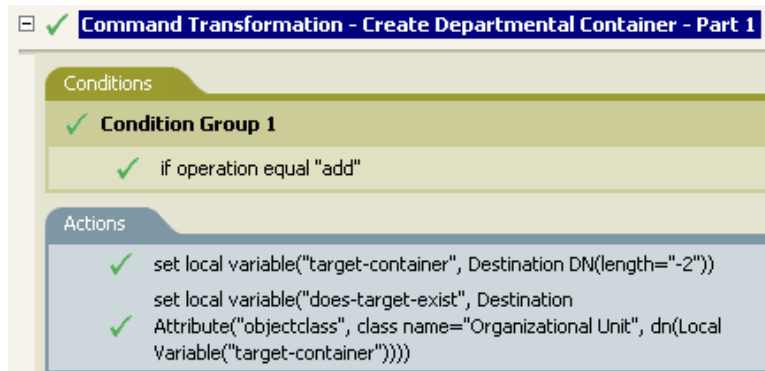
指定要包含的 RDN 數目。負數會解譯為 (段落總數 + 長度) + 1。例如，如果某個 DN 有 5 個段落，長度為 -1，則數目 = (5 + (-1)) + 1 = 5；若長度為 -2，則數目 = (5 + (-2)) + 1 = 4，以此類推。

備註

如果開始和長度設為預設值 {0,-1}，則會使用整個 DN，否則就只會使用開始和長度所指定的部份 DN。

範例

此範例使用 Destination DN 記號設定 target-container 的本地變數值。如果此容器不存在，該規則便會為「使用者」物件建立一個部門容器。該規則來自 Identity Manager 3.0 之預先定義的規則。如需相關資訊，請參閱「[指令轉換：建立部門容器：第 1 部份和第 2 部份](#)」，[第 69 頁](#)。



..... Destination DN(length="-2")

2.7.7 目的名稱

展開至目前操作中指定之目的 DN 的不合法相對可辨識名稱 (Relative Distinguished Name，RDN)。

範例

..... Destination Name()

2.7.8 授權


展開至目前物件中獲授予授權的值。

欄位

名稱

指定授權的名稱。

範例

-----  Entitlement("manager")

2.7.9 全域組態值

展開至全域組態值的值。

欄位

名稱

全域組態值的名稱。

範例

-----  Global Configuration Value("Company Name")

2.7.10 本地變數

展開至本地變數的值。

欄位

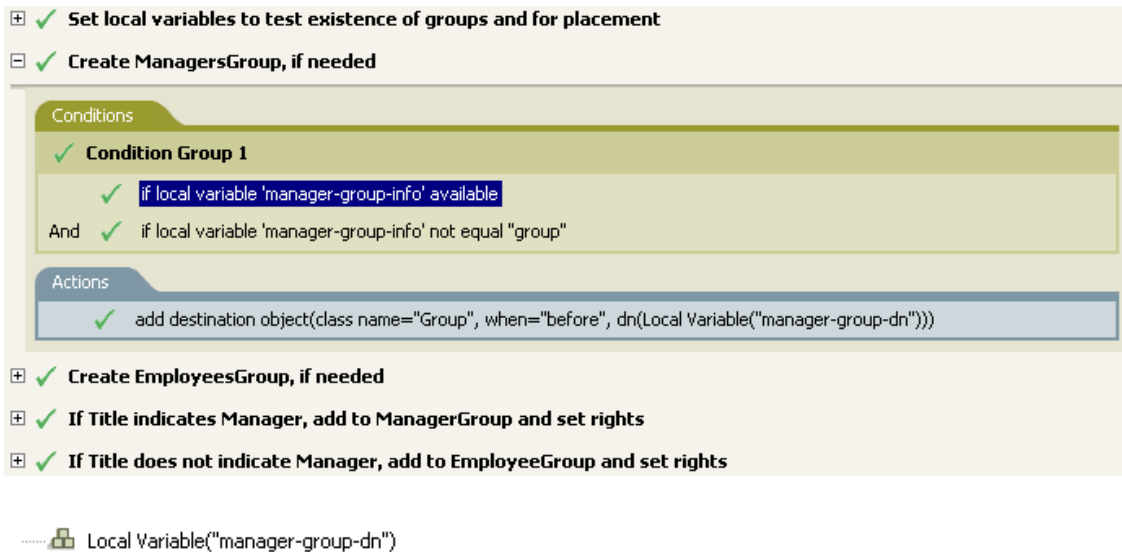
名稱

指定本地變數的名稱。

範例

此範例來自於「依 Title 管理使用者的群組」規則，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

「新增目的物件」的動作會使用「本地變數」記號。




The screenshot displays a rule configuration window with the following elements:

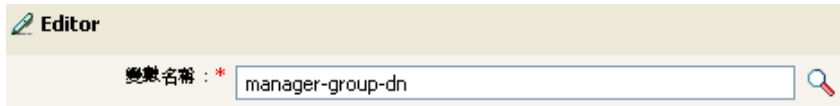
- Conditions:**
 - Condition Group 1
 - if local variable 'manager-group-info' available
 - And if local variable 'manager-group-info' not equal "group"
- Actions:**
 - add destination object(class name="Group", when="before", dn(Local Variable("manager-group-dn")))

Below the configuration window, there are several expandable sections:

- Set local variables to test existence of groups and for placement
- Create ManagersGroup, if needed
- Create EmployeesGroup, if needed
- If Title indicates Manager, add to ManagerGroup and set rights
- If Title does not indicate Manager, add to EmployeeGroup and set rights

At the bottom, there is a local variable definition:

-----  Local Variable("manager-group-dn")



「本地變數」僅有在先前已於規則中使用了「設定本地變數」動作的情形下才能使用。它會設定儲存在「本地變數」中的值。在「編輯器」中，您可以按一下瀏覽圖示和所有列出的已定義本地變數。選取正確的本地變數。

本地變數的值為 `group-manager-dn`。在此規則之前的規則中，「設定位置變數」動作將 `group-manager-dn` 定義為管理員之群組 `Users\ManagersGroup` 的 DN。

2.7.11 具名密碼


展開至驅動程式中的具名密碼。

欄位

名稱

指定密碼的名稱。

範例

.....  Named Password("password")

2.7.12 操作

展開至目前操作的名稱。

範例

.....  Operation()

2.7.13 操作屬性

展開至目前操作中的屬性值。其中不包含從修改操作中移除的值。

欄位

名稱

指定屬性的名稱。

範例

此範例有四個規則 (Rule)，這些規則根據 Surname 屬性的第一個字元實作「使用者」物件的「佈置」規則 (Policy)。可以產生追蹤訊息和自定的 Novell Audit 事件。該項規則的名稱為「依姓放置的規則」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

Setup Local Variables

Surname A-I: place in Users1

Conditions

Condition Group 1

- if class name equal "User"
- And if operation attribute 'Surname' match "[a-].*"

Actions

- set operation destination DN(dn("Training\Users\Active\Users1"+"\"+Operation Attribute("CN")))
- trace message(color="yellow", Local Variable("LVUsers1"))
- generate event(id="1000", text1=Local Variable("LVUsers1"))

Surname J-R: place in Users2

Surname S-Z: place in Users3

"Training\Users\Active\Users1"

"\"

Operation Attribute("CN")

Editor

名稱: * CN

「設定操作目的 DN」動作包含 Operation Attribute 記號。「操作屬性」記號會將「目的 DN」設定為 CN 屬性。此規則會取得 Training\Users\Active\Users 的網路位置，並新增 \ 加上 CN 屬性的值。

2.7.14 操作內容

展開至目前操作中的操作內容值。

欄位

名稱

指定操作內容的名稱


範例

.....  Operation Property("myStoredProperty")

2.7.15 密碼

展開至目前操作中的密碼。

範例

.....  Password()

2.7.16 移除的屬性


展開至目前操作中將被移除的屬性值。此屬性僅會套用至修改操作。

欄位

名稱

指定屬性的名稱

範例

.....  Removed Attribute("OU")

2.7.17 移除的授權

展開至目前操作中被撤銷的授權值。

欄位

名稱

指定授權的名稱。

範例

.....  Removed Entitlement("manager")

2.7.18 來源屬性

展開至來自來源資料儲存內物件的屬性值。

欄位


類別名稱

(選擇性) 指定目標物件的類別名稱。保留為空白以使用目前物件的類別名稱。

名稱

屬性名稱。

範例

 Source Attribute("CN", class name="User")

2.7.19 來源 DN

展開至目前操作中的來源 DN。

欄位

轉換

選取是否要將 DN 轉換為目的資料儲存所使用的格式。

開始

指定要開始處理的相對可辨識名稱 (Relative Distinguished Name, RDN) 索引：

- ◆ 索引 0 為最根部的相對可辨識名稱 (Relative Distinguished Name, RDN)
- ◆ 正索引為最根部之相對可辨識名稱 (Relative Distinguished Name, RDN) 的偏移
- ◆ 索引 -1 為最葉部的段落
- ◆ 負索引為從最葉部之相對可辨識名稱 (Relative Distinguished Name, RDN) 至最根部之相對可辨識名稱的偏移


長度

要包含的相對可辨識名稱 (Relative Distinguished Name, RDN) 段落數目。負數會解譯為 (段落總數 + 長度) + 1。例如，如果某個 DN 有 5 個段落，長度為 -1，則數字 = $(5 + (-1)) + 1 = 5$ ；若長度為 -2，則數字 = $(5 + (-2)) + 1 = 4$ ，以此類推。

備註

如果開始和長度設為預設值 {0,-1}，則會使用整個 DN，否則就只會使用開始和長度所指定的部份 DN。


範例

 Source DN()

2.7.20 來源名稱

展開至目前操作之來源 DN 的不合法相對可辨識名稱 (Relative Distinguished Name, RDN)。

範例

 Source Name()

2.7.21 文字

展開至文字。

欄位

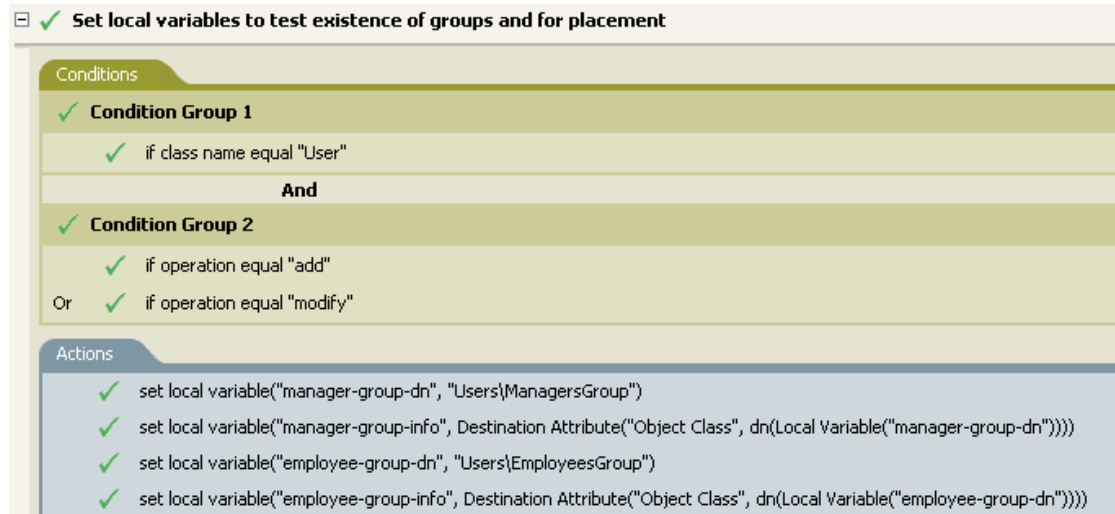
文字

指定文字。

範例

此範例來自於「依 Title 管理使用者的群組」規則，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

Text 記號在「設定位置變數」動作中用來定義管理員的群組 DN。Text 記號可包含物件或純文字。




Set local variables to test existence of groups and for placement

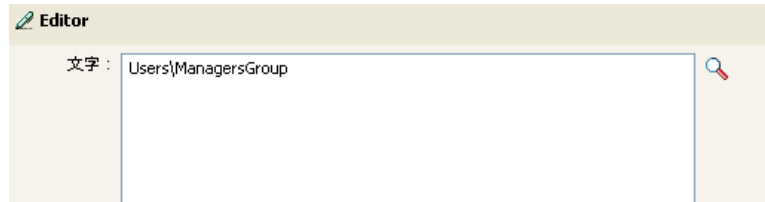
Conditions

- Condition Group 1**
 - if class name equal "User"
- And**
- Condition Group 2**
 - if operation equal "add"
 - Or if operation equal "modify"

Actions

- set local variable("manager-group-dn", "Users\ManagersGroup")
- set local variable("manager-group-info", Destination Attribute("Object Class", dn(Local Variable("manager-group-dn"))))
- set local variable("employee-group-dn", "Users\EmployeesGroup")
- set local variable("employee-group-info", Destination Attribute("Object Class", dn(Local Variable("employee-group-dn"))))

 "Users\ManagersGroup"



Editor

文字:

Text 名詞包含管理員的群組 DN。您可以瀏覽至要使用的物件，或在編輯器中輸入資訊。

2.7.22 唯一名稱

展開至模式形式的名稱；根據指定的準則，該名稱在資料儲存中為唯一名稱。

欄位

名稱

指定要檢查其獨特性的屬性名稱。

範圍

指定獨特性的檢查範圍。

開始搜尋

選取搜尋的起始點。起始點可以是資料儲存的根部，或由 DN 或關聯指定。

模式

指定要使用的模式，藉由「引數產生器」產生唯一值。

計數器啟動

指定需要尋找唯一名稱時，計數器使用的開始數字。

數位

指定計數器的數字寬度，預設值為 1。開頭核取方塊為 0 的「記事」計數器會預加 0，以符合數字長度。例如，如果數字寬度為 3，起始的唯一值便會先以 001，再以 002 附加上去，以此類推。

備註

就每個指定的模式而言，查詢的執行對象是目的資料儲存，並使用提供的屬性名稱、範圍及搜尋起始進行。每個指定的模式都會依序嘗試使用，直到找到的值未傳回任何找到的物件為止。

如果所有的指定模式都已用盡，便會附加計數器到最後一個模式，且重複嘗試使用該模式（每次增加計數器），直到查詢未傳回任何例項為止。


計數器可設為使用計數器啟動欄位，從不同的數字開始起算。計數器會使用數字欄位所指定的數字數目。如果數字的數目少於指定數目，計數器的右方便會填入零。數字的數目超過指定數目時，則不會產生任何唯一名稱，且所含規則會傳回錯誤狀態。

如果目的資料儲存為 Identity Vault 且名稱欄位保留空白，則會針對虛擬屬性 "[Entry].rdn" 執行搜尋，且該屬性不論命名屬性為何，皆代表物件的相對可辨識名稱 (Relative Distinguished Name, RDN)。如果目的資料儲存為已連接的應用程式，則需要名稱欄位。

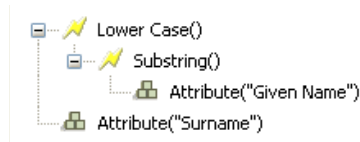
範例

 Unique Name("CN", Lower Case()+Attribute("Surname"))

以下為建構唯一名稱引數時的「編輯器」窗格範例：



為了提供唯一名稱，已建構下列模式：



如果此模式未產生唯一名稱，則會附加一個數字，並遞增至指定的數字數目。在此範例中，附加的數字會在錯誤發生前，產生九個額外的唯一名稱。

2.7.23 Unmatched Source DN

展開至目前操作中的部份來源 DN，且其對應於最近一次的「If 來源 DN」條件比對中，不符合該條件的部份 DN。

欄位

轉換

選取是否要將 DN 轉換為目的資料儲存所使用的格式。

備註

如果沒有相符項目，則會使用整個 DN。

範例

此範例取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[比對：鏡像複製訂閱者：LDAP 格式](#)」，第 85 頁。

「尋找相符物件」動作會使用 Unmatched Source DN 記號，建立輕量目錄存取協定 (Lightweight Directory Access Protocol, LDAP) 格式中的相符資訊。如此便會採用來源 DN 中不相符的部份建立相符。

Matching - Subscriber Mirrored - LDAP format

Conditions

✓ Condition Group 1

✓ if source DN in subtree "[Enter base of source hierarchy]"

Actions

✓ set local variable("dest-base", "[Enter base of destination hierarchy])"

✓ find matching object(scope="entry", dn(Unmatched Source DN(convert="true")+","+Local Variable("dest-base")))

Unmatched Source DN(convert="true")
,
Local Variable("dest-base")

Editor

轉換為目的 DN 格式: true

2.7.24 XPath

展開至 XPath 1.0 運算式的計算結果。

欄位

運算式

指定要執行的 XPath 1.0 運算式。

範例

..... XPath("//*[@attr-name='OU']//value[start-with(string(.), 'xxx')]")

2.8 動詞記號

本節包含使用「引數產生器」介面時，可用的所有動詞記號之詳細參考資訊。

- ◆ 「逸出目的 DN」，第 187 頁
- ◆ 「逸出來源 DN」，第 187 頁
- ◆ 「小寫」，第 187 頁
- ◆ 「剖析 DN」，第 188 頁
- ◆ 「取代全部」，第 190 頁
- ◆ 「取代第一個」，第 191 頁

- ◆ 「子字串」，第 192 頁
- ◆ 「大寫」，第 193 頁

2.8.1 逸出目的 DN

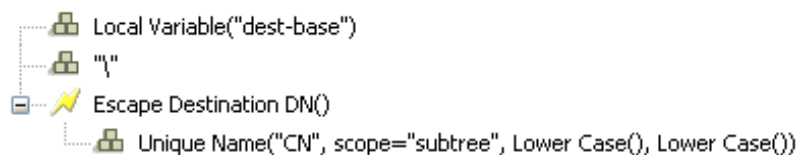
根據目的資料儲存的 DN 格式規則逸出字串。

範例

此範例取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[佈置：一般發行者](#)」，第 91 頁。

「設定目的操作 DN」動作會使用 Escape Destination DN 記號，建立「使用者」物件的目的 DN。

The screenshot shows the configuration for 'Placement - Publisher Flat'. It includes a 'Conditions' section with 'Condition Group 1' containing the rule 'if class name equal "User"'. The 'Actions' section contains two actions: 'set local variable("dest-base", "[Enter DN of destination container])' and 'set operation destination DN(dn(Local Variable("dest-base")+"\"+Escape Destination DN(Unique Name("CN", scope="subtree", Lower Case(Substring(length="1", Operation Attribute("Given Name"))+Operation Attribute("Surname")), Lower Case(Substring(length="2", Operation Attribute("Given Name"))+Operation Attribute("Surname"))))))'.



Escape Destination DN 記號會採用「唯一名稱」中的值，並將其設為目的 DN 的格式。

2.8.2 逸出來源 DN

根據來源資料儲存的 DN 格式規則逸出字串。

範例



2.8.3 小寫

將字串中的字元轉換為小寫。

範例

此範例將電子郵件地址設為 `name@slartybartfast.com`，其中名稱等於「名」的第一個字元加上「姓」。該規則的名稱為「規則：從「名」和「姓」建立電子郵件地址」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「可下載的 Identity Manager 規則」，第 32 頁。

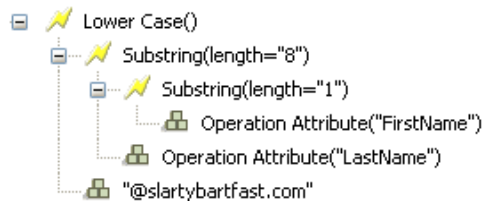
The screenshot shows a rule configuration window titled "Set email address: name@slartybartfast.com; name = (1 char of Given Name + Surname) <= 8 chars". It is divided into two main sections: "Conditions" and "Actions".

Conditions:

- Condition Group 1
 - if class name equal "User"
 - And if operation attribute 'Given Name' available
 - And if operation attribute 'Surname' available

Actions:

- strip operation attribute("Internet Email Address")
- set destination attribute value("Internet Email Address", Lower Case(Substring(length="8", Substring(length="1", Operation Attribute("FirstName"))+Operation Attribute("LastName"))+"@slartybartfast.com"))



「小寫」記號會「設定目的」動作之屬性值中的所有資訊設為小寫。

2.8.4 剖析 DN

將 DN 轉換為其他格式。

欄位

開始

指定要開始處理的相對可辨識名稱 (Relative Distinguished Name, RDN) 索引：

- ◆ 索引 0 為最根部的相對可辨識名稱 (Relative Distinguished Name, RDN)
- ◆ 正索引為最根部之相對可辨識名稱 (Relative Distinguished Name, RDN) 的偏移
- ◆ 索引 -1 為最葉部的段落
- ◆ 負索引為從最葉部之相對可辨識名稱 (Relative Distinguished Name, RDN) 至最根部之相對可辨識名稱的偏移

長度

要包含的相對可辨識名稱 (Relative Distinguished Name, RDN) 數目。負數會解譯為 (段落總數 + 長度) + 1。例如，如果某個 DN 有 5 個段落，長度為 -1，則數目 = (5 + (-1)) + 1 = 5；若長度為 -2，則數字 = (5 + (-2)) + 1 = 4，以此類推。

來源 DN 格式

指定用於解析來源 DN 的格式。

目的 DN 格式

指定用於輸出已解析之 DN 的格式。

來源 DN 分隔符

如果「來源 DN 格式」設為自定，則指定自定來源 DN 分隔符。

目的 DN 分隔符

如果「目的 DN 格式」設為自定，則指定自定目的 DN 分隔符。

備註

如果開始和長度設為預設值 {0,-1}，則會使用整個 DN，否則就只會使用開始和長度所指定的部份 DN。

指定自定 DN 格式時，組成分隔符集的八個字元定義如下：

1. 輸入的名稱布林值旗標：0 的意思是指未輸入名稱，1 的意思則是指已輸入名稱
2. Unicode 非映射字元布林值旗標：0 的意思是指不要將無法映射的 Unicode 字元解譯為逸出的十六進位數字，例如 \FEFF。eDirectory 不接受下列 Unicode 字元：0xfeff、0xffffe、0xffffd 和 0xffff。
3. 相對 RDN 分隔符
4. RDN 分隔符
5. 名稱分隔元
6. 名稱值分隔符
7. 萬用字元
8. 逸出字元

如果 RDN 分隔符和相對 RDN 分隔符是相同的字元，名稱的走向即為由根部向右，否則即為由根部向左。

如果分隔符號集中的字元超過八個，多餘的字元就會視為需要逸出，但不具有其他特殊意義的字元。

範例

該範例使用「剖析 DN」建立「新增目的屬性值」動作的值。此範例取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「指令轉換：建立部門容器：第 1 部份和第 2 部份」，第 69 頁。

Command Transformation - Create Departmental Container - Part 2

Conditions

✓ **Condition Group 1**

- ✓ if local variable 'does-target-exist' available
- And ✓ if local variable 'does-target-exist' equal ""

Actions

- ✓ add destination object(class name="organizational Unit", direct="true", dn(Local Variable("target-container")))
- ✓ add destination attribute value("ou", direct="true", dn(Local Variable("target-container")), Parse DN("dest-dn", "dot", length="1", start="-1", Local Variable("target-container")))

Parse DN("dest-dn", "dot", length="1", start="-1")

- Local Variable("target-container")

Editor

開始:

長度:

來源 DN 格式:

目的 DN 格式:

「剖析 DN」記號是採用來源 DN 中的資訊，並將其轉換為點標記。「剖析 DN」中的資訊儲存於 OU 的屬性值中。

2.8.5 取代全部

取代字串中出現的所有一般運算式。

欄位

一般運算式

指定符合要取代之子字串的一般運算式。

取代為

指定取代字串。

備註

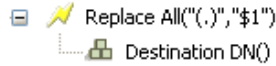
如需建立一般運算式的詳細資料，請參閱：

- ◆ Sun 的 Java 網站 (<http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html>)

- ◆ Sun 撥 Java 網站 ([http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#replaceAll \(java.lang.String\)](http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#replaceAll(java.lang.String)))

CASE_INSENSITIVE、DOTALL 和 UNICODE_CASE 模式選項都會使用，但卻可以藉由使用適當的內嵌式逸出加以反轉。

範例



2.8.6 取代第一個

取代字串中首次出現的一般運算式。

欄位

一般運算式

指定符合要取代之子字串的一般運算式。

取代為

指定取代字串。

備註

相符例項會由「取代為」欄位中指定值所指定的字串取代。

如需建立一般運算式的詳細資料，請參閱：

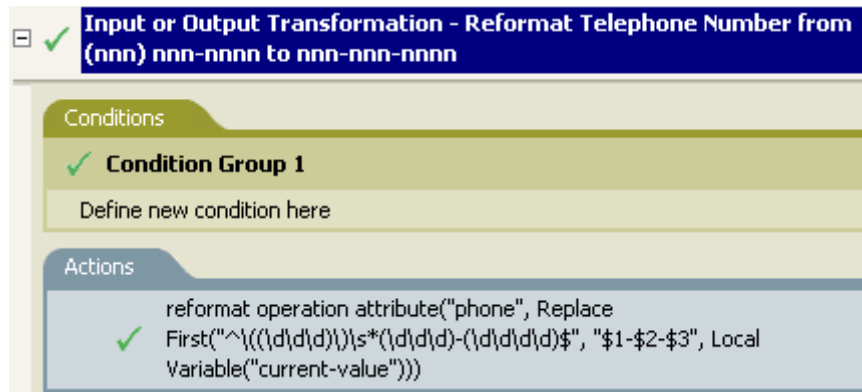
- ◆ <http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html> (<http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html>)
- ◆ [http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#replaceAll \(java.lang.String\)](http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#replaceAll(java.lang.String)) ([http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#replaceAll \(java.lang.String\)](http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#replaceAll(java.lang.String)))

CASE_INSENSITIVE、DOTALL 和 UNICODE_CASE 模式選項都會使用，但卻可以藉由使用適當的內嵌式逸出加以反轉。

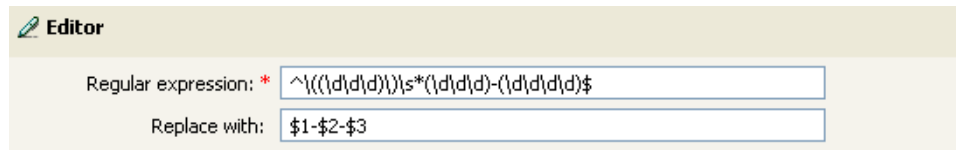
範例

該範例將電話號碼從 (nnn)-nnn-nnnn 重新格式化為 nnn-nnn-nnnn。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「輸入或輸出轉換：將電話號碼從 (nnn) nnn-nnnn 重新格式化為 nnn-nnn-nnnn」，第 81 頁。

Replace First 記號用於「重新格式化操作屬性」動作中。



⊞ ⚡ Replace First("^((\d\d\d))s*(\d\d\d)-(\d\d\d\d)\$", "\$1-\$2-\$3")
└─ Local Variable("current-value")



一般運算式 `^((\d\d\d))s*(\d\d\d)-(\d\d\d\d)$` 代表 (nnn) nnn-nnnn，而一般運算式 `$1-$2-$3` 則代表 nnn。此規則會將電話號碼的格式從 (nnn) nnn-nnnn 轉換為 nnn-xxx-xxxx。

2.8.7 子字串

擷取部份字串。

欄位

開始

指定開始字元索引：

- ◆ 索引 0 為第一個字元。
- ◆ 正索引為字串起點的偏移
- ◆ 索引 -1 為最後一個字元
- ◆ 負索引為從最後一個字元至字元起點的偏移

例如，如果起點指定為 -2，則會從結尾讀取第一個字元來起算。如果指定 -3，則會從結尾算起 2 個字元處開始。

長度

從起點開始，要包含在子字串中的字元數目。負數會解譯為 (段落總數 + 長度) + 1。例如，-1 即代表整個長度或原始字串。如果指定 -2，則長度為整個 -1。如果字串有 5 個字元，且長度為 -1，則數目 = (5 + (-1)) + 1 = 5, -2 = (5 + (-2)) + 1 = 4，以此類推。

範例

此範例將電子郵件地址設為 `name@slartybartfast.com`，其中名稱等於「名」的第一個字元加上「姓」。該規則的名稱為「規則：從「名」和「姓」建立電子郵件地址」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

Set email address: name@slartybartfast.com; name = (1 char of Given Name + Surname) <= 8 chars

Conditions

- Condition Group 1
 - if class name equal "User"
 - And if operation attribute 'Given Name' available
 - And if operation attribute 'Surname' available

Actions

- strip operation attribute("Internet Email Address")
- set destination attribute value("Internet Email Address", Lower Case(Substring(length="8", Substring(length="1", Operation Attribute("FirstName"))+Operation Attribute("LastName"))+"@slartybartfast.com"))

Lower Case()
Substring(length="8")
Substring(length="1")
Operation Attribute("FirstName")
Operation Attribute("LastName")
"@slartybartfast.com"

「子字串」記號在「設定目的屬性值」動作中會使用兩次。該記號會採用 First Name 屬性的第一個字元，並將 Last Name 屬性的八個字元一併新增，以形成一個子字串。

2.8.8 大寫

將字串中的字元轉換為大寫。

範例

此範例會將「使用者」物件的名和姓屬性轉換為大寫。規則名稱為「規則：將名 / 姓轉換為大寫」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

The screenshot shows a rule configuration window titled "Convert First/Last name to uppercase". It is divided into "Conditions" and "Actions" sections. Under "Conditions", there are two condition groups: "Condition Group 1" with the condition "if class name equal 'User'", and "Condition Group 2" which is an "Or" group containing "if operation attribute 'Given Name' changing" and "if operation attribute 'Surname' changing". Under "Actions", there are two actions: "reformat operation attribute('Given Name', Upper Case(Operation Attribute('Given Name')))" and "reformat operation attribute('Surname', Upper Case(Operation Attribute('Surname')))".

Upper Case()
└─ Operation Attribute("Given Name")

2.9 值

本節包含常見規則產生器值的清單。

2.9.1 比較模式

表格 2-7 比較模式

模式	描述
case	區分大小寫的逐字元比較。
nocase	不區分大小寫的逐字元比較。
regex	整個字串的一般運算式比對。預設為區分大小寫，但運算式中的逸出會將其變更。 請參閱 Sun 的 Java 網站 (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html) 和 Sun 的 Java 網站 (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#matches())。
	CASE_INSENSITIVE、DOTALL 和 UNICODE_CASE 模式選項都會使用，但卻可以藉由使用適當的內嵌式逸出加以反轉。
src-dn	使用適合來源資料儲存之 DN 格式的語意進行比較。
dest-dn	使用適合目的資料儲存之 DN 格式的語意進行比較。

模式	描述
numeric	以數字的方式進行比較。
octet	比較八進位 (Base64 編碼) 值。
structured	根據屬性的結構化語法比較規則，比較結構化屬性。

使用 iManager 內的規則產生器定義規則

「規則產生器」是一種完整的圖形化介面，可用於建立並管理定義已連接系統之間資料交換的規則。

這幾節接下來會提供規則和如何使用「規則產生器」的資訊：

- ◆ 「規則」，第 33 頁
- ◆ 「iManager 中的規則產生器任務」，第 198 頁

本節也包含下列的詳細參考章節：

- ◆ 「條件」，第 230 頁
- ◆ 「動作」，第 247 頁
- ◆ 「名詞記號」，第 285 頁
- ◆ 「動詞記號」，第 298 頁

3.1 規則

做為瞭解規則如何運作的一部分，您務必要瞭解規則的元件。

- ◆ 規則 (Policy) 是由規則 (Rule) 組成。
- ◆ 規則 (Rule) 則是由一組條件 (請參閱「條件」，第 230 頁) 組成，定義的動作 (請參閱「動作」，第 247 頁) 發生前必須先符合這些條件。
- ◆ 動作的動態引數可以含有衍生自執行時期所展開的記號。
- ◆ 記號可分開成爲兩個類別：名詞 (請參閱「名詞記號」，第 285 頁) 和動詞 (請參閱「動詞記號」，第 298 頁)。
 - ◆ 名詞記號會展開爲衍生自目前操作、來源或目的資料儲存，或是某外部來源的值。
 - ◆ 動詞記號則會修改從屬於該記號之其他記號的串連結果。
- ◆ 一般運算式 (請參閱「一般運算式」，第 229 頁) 和 XPath 1.0 運算式 (請參閱「XPath 1.0 運算式」，第 229 頁) 常用於規則 (Rule) 中，以建立規則 (Policy) 的所需結果。
- ◆ 規則運作於 XDS 文件上，其主要用途爲檢查並修改該文件。
- ◆ 操作就是 XDS 文件中的任何元素，該元素則爲輸入元素和輸出元素的子代。這些元素是 Novell 之 nds.dtd 的一部份；如需相關資訊，請參閱 [NDS DTD \(http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/ref/ndsdttd/index.html\)](http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/ref/ndsdttd/index.html)。
- ◆ 操作通常代表事件、指令和狀態。
- ◆ 規則會個別套用至每項操作。當規則 (Policy) 輪流套用至每項操作時，該操作即變成目前的操作。每個規則 (Rule) 會依序套用至目前的操作。所有規則 (Rule) 會套用至目前的操作，除非某動作是由使後續規則 (Rule) 不再受到套用的先前規則 (Rule) 所執行。
- ◆ 規則 (Policy) 也可以從文件外部取得其他網路位置，並造成不會反應在結果文件中的副作用。

3.2 iManager 中的規則產生器任務

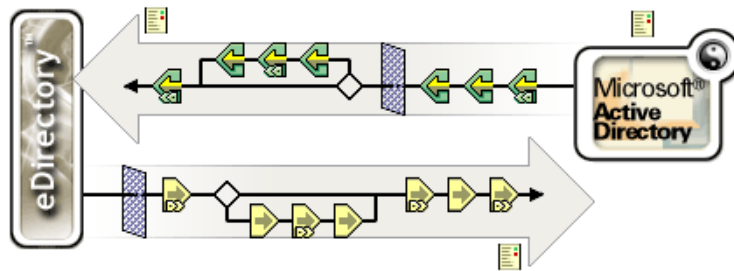
本節包含在「規則產生器」中執行常見任務的指示：

- 「開啓「規則產生器」」，第 198 頁
- 「建立規則」，第 198 頁
- 「修改規則」，第 207 頁
- 「在規則 (Policy) 內定義各別的規則 (Rule)」，第 199 頁
- 「在規則 (Rule) 內定義各別的引數」，第 200 頁
- 「使用預先定義的規則」，第 209 頁

3.2.1 開啓「規則產生器」

- 1 在 iManager 中展開「Identity Manager」角色，然後按一下「Identity Manager 概觀」。
- 2 指定驅動程式集。
- 3 按一下要管理規則的驅動程式。此時會開啓「Identity Manager 驅動程式概觀」：

特性 3-1 Identity Manager 驅動程式概觀



您可以從「Identity Manager 驅動程式概觀」中管理規則。

3.2.2 建立規則

- 1 開啓想要管理之驅動程式的「Identity Manager 驅動程式概觀」。
- 2 按一下代表要定義規則的圖示。
 - ◻ 代表一個未定義的規則。
 - ◀▶ 代表一個已定義的規則。
- 3 按一下「插入」。
- 4 為新規則輸入名稱，然後選取「規則產生器」。
- 5 此時會顯示該項規則。若要為此規則 (Policy) 定義一個或多個規則 (Rule)，請按一下「附加新規則」，然後遵循「在規則 (Policy) 內定義各別的規則 (Rule)」，第 199 頁中的指示進行。

3.2.3 在規則 (Policy) 內定義各別的規則 (Rule)

規則 (Rule) 可在「規則 (Policy) 產生器」的「規則 (Rule) 產生器」中定義。

特性 3-2 「規則 (Policy) 產生器」的「規則 (Rule) 產生器」視窗



「規則 (Rule) 產生器」介面可讓您使用智慧型下拉式功能表快速地建立及修改規則。

在「規則產生器」中，您可以定義一組條件，定義的動作發生前必須先符合這些條件。

例如，若您需要建立不允許任何新物件新增至您的環境的規則，則可以定義類似下列的規則：在發生新增操作時否決該項操作。

若要在「規則產生器」中實作此邏輯，您可以選取下列條件：

特性 3-3 在規則產生器介面中移動使用者條件



以及下列動作：

特性 3-4 在規則產生器介面中否決動作



如需「規則產生器」內可用的條件和動作的詳細參考資訊，請參閱「條件」，第 230 頁和「動作」，第 247 頁。









祕訣

若要建立更複雜的條件，您可以加入條件和結合 **and/or** 陳述式的條件群組。您可以藉由選取條件結構修改這些加入的條件：

特性 **3-5** 條件結構選項圓鈕

選取條件結構：

- 或 條件, 且 群組
- 且 條件, 或 群組

- ◆ 按一下  圖示以查看欄位的值清單。在上述範例中，此圖示會開啓一個有效類別名稱的清單。
- ◆ 按一下  圖示，以使用「引數產生器」介面來建構引數。
- ◆ 按一下  圖示來顯示規則 (Policy)、規則 (Rule)、條件或動作。按一下  圖示重新啓用它。
- ◆ 按一下  圖示新增備註到規則 (Policy) 或規則 (Rule) 中。備註會直接儲存在規則 (Policy) 或規則 (Rule) 上，並且可以視所需時間的長短保存。
- ◆ 使用「剪下」/「複製」/「貼上」圖示， 來使用「規則產生器」的剪貼簿。若剪貼簿上目前的內容在該位置上無效，則「貼上」圖示會被停用。
- ◆ 使用  圖示來新增、移除和放置條件。
- ◆ 使用 **Append Condition Group** 按鈕來新增條件群組。
- ◆ 使用  圖示來移除及放置條件群組。

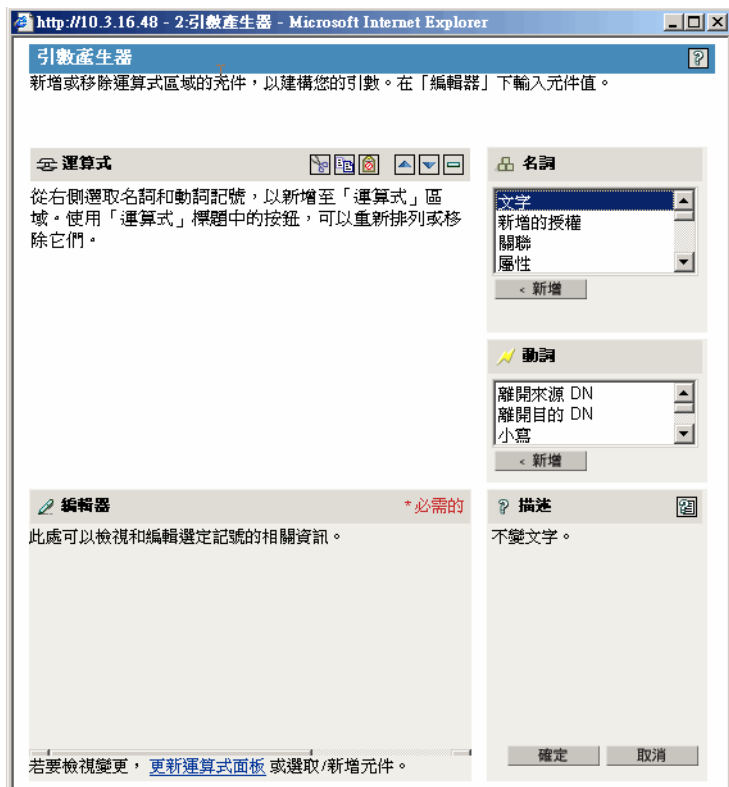
3.2.4 在規則 (Rule) 內定義各別的引數

「引數產生器」提供動態的圖形化介面，可讓您建構「規則產生器」內使用之複雜的引數運算式。若要存取「引數產生器」，請參閱「[引數產生器](#)」，第 203 頁。

引數由動作動態使用，並且可以衍生自執行時期所展開的記號。

記號可分開成爲兩個類別：名詞和動詞。名詞記號會展開爲衍生自目前操作、來源或目的資料儲存，或是某外部來源的值。動詞記號則會修改從屬於該記號之其他記號的串連結果。

特性 3-6 預設的引數產生器介面



若要定義運算式，請選取一或多個名詞記號 (值、物件、變數等)，並結合動詞記號 (子字串、逸出、大寫和小寫)，以建構引數。結合多個記號可建構複雜的引數。

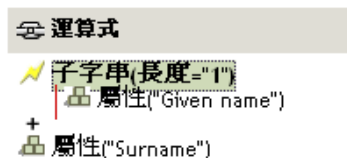
例如，若要將引數設定爲某個屬性值，可以選取屬性記號，然後選取屬性名稱。

特性 3-7 編輯器將 *ds.novell* 顯示爲文字引數



若您僅想要使用此屬性的一部份，可以結合屬性記號與子字串記號：



特性 3-8 運算式在 *Given Name* 屬性上顯示「長度 1 的子字串」，並結合 *Surname* 屬性。



在新增記號後，您就可以在編輯器內編輯欄位。

如需「引數產生器」內可用名詞和動詞的詳細參考資訊，請參閱「名詞記號」，第 285 頁和「動詞記號」，第 298 頁。

祕訣

- ◆ 若要建立更複雜的條件，您可以加入條件或結合 **and/or** 陳述式的條件群組。
- ◆ 使用  圖示來移動及刪除名詞記號和動詞記號。
- ◆ 按一下  圖示以查看欄位的值清單。
- ◆ 在新增名詞記號或動詞記號後，您可以在編輯器中提供值，然後立即新增其他的名詞記號或動詞記號。您不需要重新整理「運算式」窗格以套用變更；它們會在執行下一個操作時出現。

儘管您會使用「引數產生器」定義大多數的引數，但還有幾個產生器會在「規則產生器」內的「條件編輯器」和「動作編輯器」中使用。每個產生器可以遞迴方式呼叫下列清單內的任一個產生器：

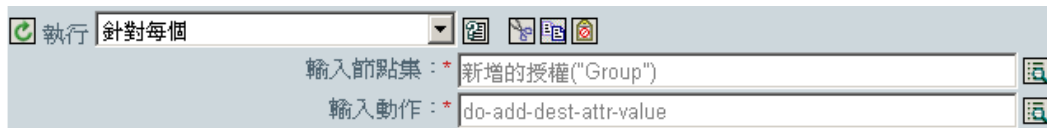
- ◆ 「引數動作產生器」，第 202 頁
- ◆ 「引數產生器」，第 203 頁
- ◆ 「相符屬性產生器」，第 204 頁
- ◆ 「動作引數元件產生器」，第 205 頁
- ◆ 「引數值清單產生器」，第 205 頁
- ◆ 「具名字串產生器」，第 206 頁
- ◆ 「條件引數元件產生器」，第 206 頁

引數動作產生器

「引數動作產生器」可讓您設定針對每個（第 259 頁）動作和實作授權（第 262 頁）動作所需的動作。

在下列範例中新增的目的屬性值動作會由將被新增至目前操作的每種「群組」授權所執行。

特性 3-9 引數動作產生器

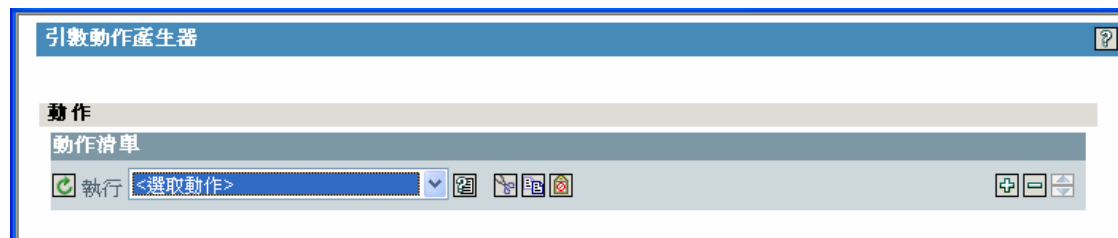


若要定義新增目的屬性值的動作，請按一下啟動「引數動作產生器」的圖示。在「引數動作產生器」中，您可以定義所需的動作。在下列範例中，將為每個新增的「群組」授權新增成員屬性至目的物件。

特性 3-10 引數動作產生器



特性 3-11 引數動作產生器



引數產生器

請按一下「編輯引數」圖示，從下列動作啟動「引數產生器」。

- ◆ 新增關聯 (第 248 頁)
- ◆ 新增目的屬性值 (第 249 頁)
- ◆ 新增目的物件 (第 250 頁)
- ◆ 新增來源屬性值 (第 251 頁)
- ◆ 附加 XML 文字 (第 253 頁)
- ◆ 清除目的屬性值 (第 254 頁) 當選取物件為 DN 或「關聯」時。
- ◆ 清除來源屬性值 (第 256 頁) 當選取物件為 DN 或「關聯」時。
- ◆ 刪除目的物件 (第 257 頁) 當選取物件為 DN 或「關聯」時。
- ◆ 刪除來源物件 (第 258 頁) 當選取物件為 DN 或「關聯」時。
- ◆ 尋找相符物件 (第 258 頁)
- ◆ 針對每個 (第 259 頁)
- ◆ 移動目的物件 (第 263 頁)
- ◆ 移動來源物件 (第 264 頁)
- ◆ 重新格式化操作屬性 (第 265 頁)
- ◆ 移除關聯 (第 265 頁)

- ◆ 移除目的屬性值 (第 266 頁)
- ◆ 移除來源屬性值 (第 267 頁)
- ◆ 重新命名目的物件 (第 267 頁) 當選取物件為 DN 或「關聯」和「輸入字串」時。
- ◆ 重新命名來源物件 (第 268 頁) 當選取物件為 DN 或「關聯」和「輸入字串」時。
- ◆ 設定目的屬性值 (第 272 頁) 當選取物件為 DN 或「關聯」和「輸入值類型」未被結構化時。
- ◆ 設定目的密碼 (第 273 頁)
- ◆ 設定本地變數 (第 274 頁)
- ◆ 設定操作關聯 (第 275 頁)
- ◆ 設定操作類別名稱 (第 275 頁)
- ◆ 設定操作目的 DN (第 275 頁)
- ◆ 設定操作內容 (第 276 頁)
- ◆ 設定操作來源 DN (第 276 頁)
- ◆ 設定操作範本 DN (第 277 頁)
- ◆ 設定來源屬性值 (第 277 頁)
- ◆ 設定來源密碼 (第 278 頁)
- ◆ 設定 XML 屬性 (第 280 頁)
- ◆ 狀態 (第 282 頁)
- ◆ 追蹤訊息 (第 283 頁)

相符屬性產生器

「相符屬性產生器」可讓您選取「尋找相符物件」，第 258 頁動作所使用的屬性及值，以判定是否符合存在於資料儲存內的物件。

例如，若您想要獲得與公用名稱和位置相符的使用者，可以選取下列條件：

特性 3-12 尋找相符物件

執行 尋找相符物件

選取範圍：子網路樹

輸入 DN："User"+屬性("UO")

輸入相符屬性：CN,L

接著您可以按一下「輸入相符屬性」欄位旁的「編輯引數」圖示，以啟動「相符屬性產生器」介面：

特性 3-13 相符屬性產生器

相符屬性

<input type="checkbox"/> 名稱：* CN	目前物件的值
<input type="checkbox"/> 名稱：* L	目前物件的值

選取「瀏覽屬性」以瀏覽至要比對的屬性並加以選取。在此例中為 L 和 CN。

第二欄可讓您藉由選取「使用目前物件的值」來比對目前儲存在屬性內的值。您可以選取「其他值」，針對其他值進行比對。您可以建立任何相要比對的值。選取值類型，並可透過「輸入狀態」欄位選取適當的產生器。

動作引數元件產生器

請在「輸入值類型」選項為「結構化」選項時選取下列動作，以啟動「動作引數元件產生器」。

- ◆ 新增目的屬性值 (第 249 頁)
- ◆ 新增來源屬性值 (第 251 頁)
- ◆ 重新格式化操作屬性 (第 265 頁)
- ◆ 移除目的屬性值 (第 266 頁)
- ◆ 移除來源屬性值 (第 267 頁)
- ◆ 設定預設的屬性值 (第 271 頁)
- ◆ 設定來源屬性值 (第 277 頁)

特性 3-14 動作引數元件產生器

動作清單

執行 新增目的屬性值

輸入屬性名稱：* Given name

輸入類別名稱：User

選取模式：直接寫入至目的資料儲存

選取物件：目前物件

輸入值類型：structured

輸入元件：* user

特性 3-15 動作引數元件產生器

引數元件產生器

當 <arg-value> 的類型屬性為 [structured] 時，引數元件會提供所含 <arg-value> 之元件的值。會評估每個所含記號，且會串連結果字串值以形成元件的值。元件的名稱由名稱屬性指定。

* 必需的

附加新元件 移除

引數元件

名稱：*	記號：*

引數值清單產生器

「引數值清單產生器」可讓您為 設定預設的屬性值 (第 271 頁) 動作建構預設的引數值。

例如，若您想要設定「不明」的預設位置，可以選取下列動作：

特性 3-16 引數值清單產生器

引數值	
<input type="checkbox"/> 類型：*	string
<input type="checkbox"/> 輸入字串：*	"Digital Airline Inc"

接著您可以按一下「輸入值」欄位旁的圖示，以啟動「引數值清單產生器」介面，並建構與以下類似的引數：

特性 3-17 引數值清單產生器

引數值	
<input type="checkbox"/> 類型：*	string
<input type="checkbox"/> 輸入字串：*	"Unknown"

具名字串產生器

「具名字串產生器」可讓您建構在某些動作中使用的 name/value 對，例如，[產生事件 \(第 260 頁\)](#)、[傳送電子郵件 \(第 269 頁\)](#) 和 [從範本傳送電子郵件 \(第 270 頁\)](#)。

對於「產生事件」動作，其具名字串對應到您可以事件提供的自定值欄位：

特性 3-18 具名字串產生器

字串	
<input type="checkbox"/> 名稱：*	to
<input type="checkbox"/> 字串值：*	"to_user1@company.com"
<input type="checkbox"/> 名稱：*	to
<input type="checkbox"/> 字串值：*	"to_user@company.com"
<input type="checkbox"/> 名稱：*	cc
<input type="checkbox"/> 字串值：*	"cc_user@company.com"
<input type="checkbox"/> 名稱：*	bcc
<input type="checkbox"/> 字串值：*	"bcc_user@company.com"
<input type="checkbox"/> 名稱：*	from
<input type="checkbox"/> 字串值：*	"from_user@company.com"
<input type="checkbox"/> 名稱：*	subject
<input type="checkbox"/> 字串值：*	"This is the e-mail subject"
<input type="checkbox"/> 名稱：*	message
<input type="checkbox"/> 字串值：*	"This is the e-mail body"

對於「傳送郵件」動作，其具名字串會對應至電子郵件的元素：

特性 3-19 傳送郵件動作

字串	
<input type="checkbox"/> 名稱：*	manager
<input type="checkbox"/> 字串值：*	"Bill Jones"
<input type="checkbox"/> 名稱：*	surname
<input type="checkbox"/> 字串值：*	"Smith"
<input type="checkbox"/> 名稱：*	given-name
<input type="checkbox"/> 字串值：*	"Joe"
<input type="checkbox"/> 名稱：*	to
<input type="checkbox"/> 字串值：*	"to_user@company.com"
<input type="checkbox"/> 名稱：*	cc
<input type="checkbox"/> 字串值：*	"cc_user@company.com"

應至啟動「具名字串產生器」動作的說明檔案包含可能的值之完整清單。

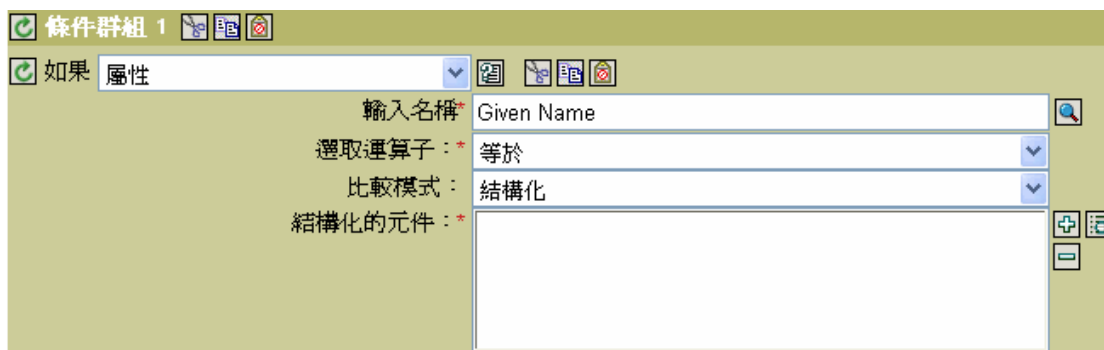
條件引數元件產生器

按一下「編輯引數」圖示即可啟動「條件引數元件產生器」。

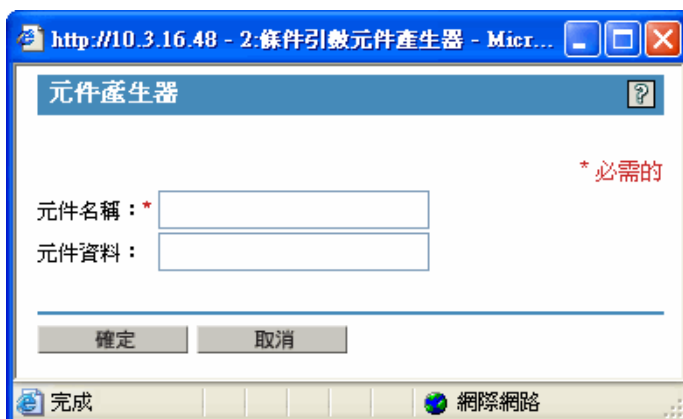
爲了查看圖示，您必須爲「模式」選取具下列條件的「結構化」選項：

- ◆ If Attribute (第 231 頁)
- ◆ If Destination Attribute (第 233 頁)
- ◆ If Source Attribute (第 243 頁)

特性 3-20 結構化選項



特性 3-21 條件引數元件產生器



3.2.5 修改規則

- 1 開啓想要管理之驅動程式的「Identity Manager 驅動程式概觀」。
- 2 按一下代表要修改規則的圖示。
- 3 選取要修改的規則，然後按一下「編輯」。


3.2.6 移除規則

從選定「規則集」中移除規則，但不將其刪除。

- 1 開啓想要管理之驅動程式的「Identity Manager 驅動程式概觀」。
- 2 按一下代表要移除規則的圖示。

若要檢視未與規則集關聯的規則：

- 1 開啓想要管理之驅動程式的「Identity Manager 驅動程式概觀」。

- 2 按一下「檢視所有規則圖示」。

若要將已移除的規則重新加入規則集：

- 1 開啓想要管理之驅動程式的「*Identity Manager* 驅動程式概觀」。
- 2 按一下要新增規則的規則集。
- 3 按一下「插入」。
- 4 選取「使用現有規則」，然後按一下瀏覽按鈕。
- 5 瀏覽至要新增的規則。

提示：確定您位於適當的容器，以查看規則。

- 6 按一下「確定」。
- 7 按一下「關閉」。

3.2.7 重新命名規則

- 1 開啓想要管理之驅動程式的「*Identity Manager* 驅動程式概觀」。
- 2 按一下代表要重新命名規則的圖示。
- 3 按一下「重新命名」並為規則重新命名。
- 4 按一下「確定」。
- 5 按一下「關閉」。

3.2.8 刪除規則

- 1 開啓想要管理之驅動程式的「*Identity Manager* 驅動程式概觀」。
- 2 按一下代表要刪除規則的圖示。
- 3 選取要刪除的規則，然後按一下「刪除」。

3.2.9 從 XML 檔案輸入規則

- 1 開啓想要管理之驅動程式的「*Identity Manager* 驅動程式概觀」。
- 2 按一下代表要輸入規則的圖示。
- 3 選取規則，然後按一下「編輯」。
- 4 按一下「插入」按鈕，然後選取「輸入包含「*DirXML*® 程序檔」的 XML 檔案」。
- 5 瀏覽至要輸入的規則檔並加以選取，然後按一下「確定」。

3.2.10 將規則輸入至 XML 檔案

- 1 開啓想要管理之驅動程式的「*Identity Manager* 驅動程式概觀」。
- 2 按一下代表要輸出規則的圖示。
- 3 選取規則，然後按一下「編輯」。
- 4 按一下「另存新檔」按鈕，然後選取要儲存 *DirXML* 程序檔之 XML 檔的位置。
- 5 按一下「儲存」。

3.2.11 建立規則參考

規則參考可讓您建立單一規則，並在多個位置參考該規則。若您擁有由一個以上的驅動程式或規則使用的規則，請建立一個參考，以簡化此規則的管理。

- 1 開啓想要管理之驅動程式的「*Identity Manager* 驅動程式概觀」。
- 2 按一下代表要新增為參考之規則的圖示。
- 3 選取規則，然後按一下「編輯」。
- 4 按一下「插入」按鈕，並選取「附加包含「*DirXML* 程序檔」之規則的參考」。
- 5 瀏覽至要參考的規則物件並加以選取，然後按一下「確定」。

3.2.12 使用預先定義的規則

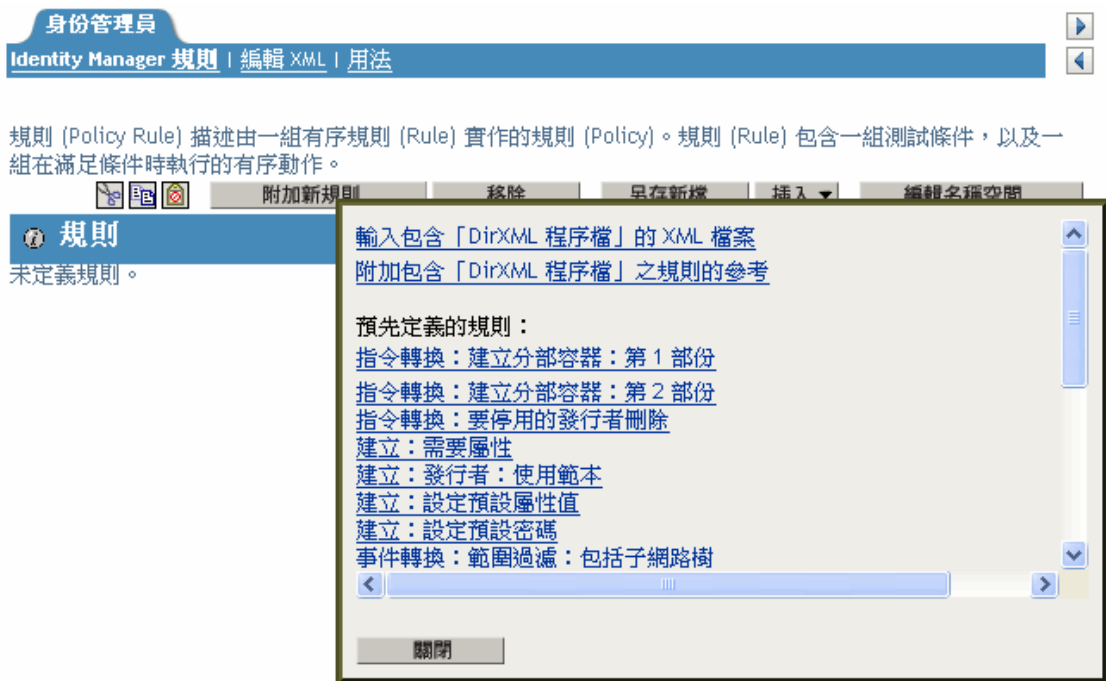
iManager 包含二十個預先定義的規則。您可以輸入及使用這些規則，並建立自己的規則。這些規則包含管理員常用的任務。您必須提供環境特定的資訊，以藉此自定規則。

- ◆ 「指令轉換：建立部門容器：第 1 部份和第 2 部份」，第 210 頁
- ◆ 「指令轉換：要停用的發行者刪除」，第 212 頁
- ◆ 「建立：需要屬性」，第 212 頁
- ◆ 「建立：發行者：使用範本」，第 213 頁
- ◆ 「建立：設定預設屬性值」，第 214 頁
- ◆ 「建立：設定預設密碼」，第 215 頁
- ◆ 「事件轉換：範圍過濾：包括子網路樹」，第 216 頁
- ◆ 「事件轉換：範圍過濾：排除子網路樹」，第 217 頁
- ◆ 「輸入或輸出轉換：將「電話號碼」從 (nnn) nnn-nnnn 重新格式化為 nnn-nnn-nnnn」，第 218 頁
- ◆ 「輸入或輸出轉換：將電話號碼從 nnn-nnn-nnnn 重新格式化為 (nnn) nnn-nnnn」，第 219 頁
- ◆ 「比對：鏡像複製發行者」，第 219 頁
- ◆ 「比對：鏡像複製訂閱者：LDAP 格式」，第 220 頁
- ◆ 「比對：依屬性值」，第 221 頁
- ◆ 「佈置：鏡像複製發行者」，第 222 頁
- ◆ 「佈置：鏡像複製訂閱者：LDAP 格式」，第 223 頁
- ◆ 「佈置：一般發行者」，第 224 頁
- ◆ 「佈置：一般訂閱者：LDAP 格式」，第 225 頁
- ◆ 「佈置：依部門的發行者」，第 226 頁
- ◆ 「佈置：依部門的訂閱者：LDAP 格式」，第 227 頁

若要存取預先定義的規則：

- 1 開啓想要管理之驅動程式的「*Identity Manager* 驅動程式概觀」。
- 2 按一下代表要新增預先定義規則 (Rule) 的規則 (Policy) 圖示。
- 3 選取規則，然後按一下「編輯」。

4 按一下「插入」，並選取要使用的預先定義規則。



指令轉換：建立部門容器：第 1 部份和第 2 部份

在目的資料儲存中建立部門容器 (若其中不存在該容器)。您可以在驅動程式內的「訂閱者指令轉換」規則 (Policy) 或「發行者指令轉換」規則 (Policy) 上實作規則 (Rule)。

使用預先定義的規則包含兩個步驟：在「指令轉換」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「指令轉換」規則 (Policy)，請跳至「輸入預先定義的規則」，第 210 頁。

建立規則

- 1 開啟想要管理之驅動程式的「Identity Manager 驅動程式概觀」。
- 2 按一下「發行者」或「訂閱者」通道上的「指令轉換規則」物件。
- 3 按一下「插入」。
- 4 為規則命名，請確定使用「規則產生器」實作規則，然後按一下「確定」。

此時會啟動「規則產生器」。

輸入預先定義的規則

- 1 在「規則產生器」中，按一下「插入」。
- 2 選取「指令轉換：建立部門容器：第 1 部份」。
- 3 按一下「插入」。
- 4 選取「指令轉換：建立部門容器：第 2 部份」。
- 5 按一下「確定」。

您環境特有的規則中沒有要變更的資訊。

Command Transformation - Create Departmental Container - Part 1

條件

如果操作等於 add

動作

```
set local variable("target-container",Destination DN(length="-2"))
set local variable("does-target-exist",Destination Attribute("objectclass",class
name="OrganizationalUnit",dn(Local Variable("target-container"))))
```

Command Transformation - Create Departmental Container - Part2

條件

如果本地變數 does-target-exist 可供使用
且 如果本地變數 does-target-exist 等於 ""

動作

```
add destination object(class name="organizationalUnit",direct="true",dn(Local Variable("target-
container")))
add destination attribute value("ou",direct="true",dn(Local Variable("target-container")),Parse
DN("dest-dn","dot",length="1",Local Variable("target-container")))
```

重要：請確定規則依順序列出。第 1 部份必須在第 2 部份之前執行。

規則內邏輯的運作方式

當某個物件的目的位置不存在時，可以使用此規則。在因為無法放置物件而遭到否決時，此規則會建立容器並將物件放置到容器內。

第 1 部份會尋找任何「新增」操作。當發生「新增」操作時，會設定兩個本地變數。第一個本地變數會被命名為 `target-container`。`target-container` 的值會被設為目的 DN。第二個本地變數會被命名為 `does-target-exist`。`does-target-exist` 的值會被設為 `objectclass` 的目的屬性值。類別會被設定為 `OrganizationalUnit`。`OrganizationalUnit` 的 DN 會被設定為 `target-container` 的本地變數。

特性 3-22 建立容器

編輯器

名稱：*

類別名稱：

選取物件：*

第 2 部份會檢查是否可以使用本地變數 `does-target-exist`。第 2 部份同時也會檢查本地變數 `does-target-exist` 的值是否設定為空白值。若為空白值，則會建立一個「組織單位」物件。組織單位的 DN 會被設定為本地變數 `target-container` 的值。同時也會新增 OU 屬性的值。OU 屬性的值會設定為新組織單位的名稱，其值藉由剖析本地變數 `target-container` 而獲得。

指令轉換：要停用的發行者刪除

將「使用者」物件的「刪除」操作轉換為在 eDirectory™ 內停用目標「使用者」物件的「修改」操作。在驅動程式內的「發行者指令轉換」規則 (Policy) 上實作規則 (Rule)。

使用預先定義的規則包含兩個步驟：在「指令轉換」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「指令轉換」規則 (Policy)，請跳至[輸入預先定義的規則 \(第 212 頁\)](#)。

建立規則

- 1 開啟想要管理之驅動程式的「Identity Manager 驅動程式概觀」。
- 2 按一下「發行者」通道上的「指令轉換規則」物件。
- 3 按一下「插入」。
- 4 為規則命名，請確定使用「規則產生器」實作規則，然後按一下「確定」。

此時會啟動「規則產生器」。

輸入預先定義的規則

- 1 在「規則產生器」中，按一下「插入」。
- 2 選取「指令轉換：要停用的發行者刪除」。
- 3 按一下「確定」。

您環境特有的規則中沒有要變更的資訊。

指令轉換：要停用的發行者刪除

條件

- 如果操作等於 delete
- 或 如果類別名稱等於 User

動作

- set destination attribute value("Login Disabled","true")
- remove association(association(Association()))

規則內邏輯的運作方式

此規則會在「刪除」指定將被傳送至 Identity Vault 時使用，通常用於回應發生在已連接系統內的「刪除」事件。以停用「使用者」物件的方式取代刪除 Identity Vault 內的「使用者」物件。在為「使用者」物件處理「刪除」指令時，「禁止登入」的目的屬性值會被設定為 true、與「使用者」物件的關聯會被移除，並且「刪除」指令會被否決。「使用者」物件無法再登入至 Novell eDirectory 網路樹，但「使用者」物件未被刪除。

建立：需要屬性

除非已填入必要的屬性，否則將不允許建立「使用者」。您可以在驅動程式內的「訂閱者建立」規則 (Policy) 或「發行者建立」規則 (Policy) 上實作規則 (Rule)。

使用預先定義的規則包含兩個步驟：在「建立」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「建立」規則 (Policy)，請跳至[「輸入預先定義的規則」，第 213 頁](#)。

建立規則

- 1 開啓想要管理之驅動程式的「*Identity Manager* 驅動程式概觀」。
- 2 按一下「發行者」或「訂閱者」通道上的「建立規則」物件。
- 3 按一下「插入」。
- 4 爲規則命名，請確定使用「規則產生器」實作規則，然後按一下「確定」。

此時會啓動「規則產生器」。

輸入預先定義的規則

- 1 在「規則產生器」中，按一下「插入」。
- 2 選取「建立：需要屬性」。
- 3 按一下「規則產生器」內的「建立：需要屬性」，以編輯規則。
- 4 從「輸入名稱欄位」刪除「[輸入必要屬性的 DN]」。
- 5 按一下瀏覽圖示，然後瀏覽至要建立之「使用者」物件所需的屬性並加以選取。
- 6 (選擇性) 若您需要一個以上的必要屬性，請按一下「加號圖示」，以新增動作。
- 7 選取「如果操作屬性無法使用則否決」，並瀏覽至其他必要的屬性。
- 8 按一下「確定」。



規則內邏輯的運作方式

此規則會在您的企業流程在目的「使用者」物件可以建立之前，於來源「使用者」物件需要使用者填入特定屬性時使用。在來源資料儲存建立「使用者」物件時，若在建立「使用者」物件時之前未提供必要屬性，則此規則會否則目的資料儲存內物件的建立程序。您可以擁有一或多個必要的屬性。

建立：發行者：使用範本

允許在建立「使用者」物件期間使用 Novell eDirectory 範本物件。在驅動程式內的「發行者建立」規則 (Policy) 上實作規則 (Rule)。

使用預先定義的規則包含兩個步驟：在「建立」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「建立」規則 (Policy)，請跳至「[輸入預先定義的規則](#)」，第 214 頁。

建立規則

- 1 開啓想要管理之驅動程式的「*Identity Manager* 驅動程式概觀」。
- 2 按一下「發行者」或「訂閱者」通道上的「建立規則」物件。
- 3 按一下「插入」。
- 4 爲規則命名，請確定使用「規則產生器」實作規則，然後按一下「確定」。

此時會啟動「規則產生器」。

輸入預先定義的規則

- 1 在「規則產生器」中，按一下「插入」。
- 2 選取「建立：發行者：使用範本」。
- 3 按一下「規則產生器」內的「建立：發行者：使用範本」，以編輯規則。
- 4 從「輸入 DN 欄位」中刪除「[輸入範本物件的 DN]」。
- 5 按一下「編輯引數圖示」以啟動「引數產生器」。
- 6 選取「名詞」清單中選取「文字」，然後按一下「新增」。
- 7 在「編輯器」中按一下瀏覽圖示，瀏覽至範本物件並加以選取，然後按一下「確定」。
- 8 按一下「確定」。



規則內邏輯的運作方式

當您要在 Identity Vault 內依範本物件來建立使用者時，可以使用此規則。若您有多個使用者擁有相同的屬性，可使用範本來節省時間。您必須在範本物件內填入資訊，並且在「使用者」物件被建立時，Identity Manager 會使用來自範本的屬性值來建立「使用者」物件。

在「使用者」物件的建立期間，規則會執行設定操作範本 DN 的動作，該動作會指示 Identity Manager 在建立物件時使用參考的範本。

建立：設定預設屬性值

可讓您在建立「使用者」物件期間設定指定屬性的預設值。您可以在驅動程式內的「訂閱者建立」規則 (Policy) 或「發行者建立」規則 (Policy) 上實作規則 (Rule)。

使用預先定義的規則包含兩個步驟：在「建立」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「建立」規則 (Policy)，請跳至「[輸入預先定義的規則](#)」，第 214 頁。

建立規則

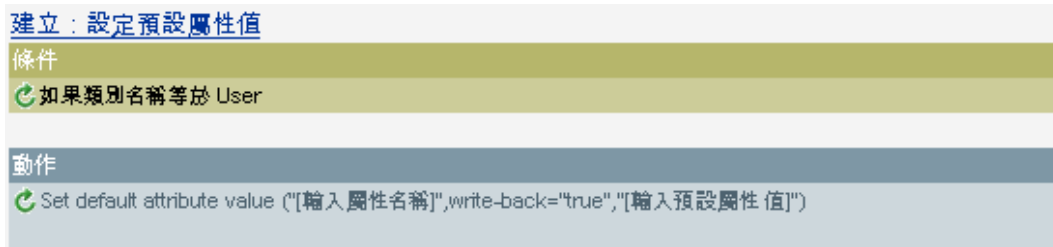
- 1 開啓想要管理之驅動程式的「Identity Manager 驅動程式概觀」。
- 2 按一下「發行者」或「訂閱者」通道上的「建立規則」物件。
- 3 按一下「插入」。
- 4 為規則命名，請確定使用「規則產生器」實作規則，然後按一下「確定」。

此時會啟動「規則產生器」。

輸入預先定義的規則

- 1 在「規則產生器」中，按一下「插入」。

- 2 選取「建立：設定預設屬性值」。
- 3 按一下「規則產生器」內的「設定預設屬性值」，以編輯規則。
- 4 從「輸入屬性名稱欄位」中刪除「[輸入屬性名稱]」。
- 5 按一下瀏覽圖示，然後瀏覽至想要建立的屬性並加以選取。
- 6 從「輸入引數值欄位」中刪除「[輸入預設屬性值]」。
- 7 按一下「編輯引數圖示」以啟動「引數值清單產生器」。
- 8 選取要為值設定的資料類型。
- 9 按一下「編輯引數」圖示以啟動「引數產生器」。
- 10 在「引數產生器」中為所需屬性建立值，然後按一下「確定」。
- 11 按一下「確定」。



規則內邏輯的運作方式

如果您要在建立「使用者」物件時填入預設的屬性值，可以使用此規則。在建立「使用者」物件時，如果且僅有在來源物件未為屬性提供值時，此規則會新增指定的屬性值。

若您想要定義一個以上的屬性值，請在動作上按一下滑鼠右鍵，然後按一下「新增」>「動作」。選取動作、設定預設的屬性值，並遵循上述步驟指定屬性值。

建立：設定預設密碼

在建立「使用者」物件期間為「使用者」物件設定預設的密碼。您可以在驅動程式內的「訂閱者建立」規則 (Policy) 或「發行者建立」規則 (Policy) 上實作規則 (Rule)。

使用預先定義的規則包含兩個步驟：在「建立」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「建立」規則 (Policy)，請跳至「輸入預先定義的規則」，第 215 頁。

建立規則

- 1 開啟想要管理之驅動程式的「Identity Manager 驅動程式概觀」。
- 2 按一下「發行者」或「訂閱者」通道上的「建立規則」物件。
- 3 按一下「插入」。
- 4 為規則命名，請確定使用「規則產生器」實作規則，然後按一下「確定」。

此時會啟動「規則產生器」。

輸入預先定義的規則

- 1 在「規則產生器」中，按一下「插入」。
- 2 選取「建立：設定預設密碼」。

3 按一下「確定」。

您環境特有的規則中沒有要變更的資訊。

建立：設定預設密碼

條件
如果類別名稱等於 User
動作
set destination password(Attributer("Given Name")+Attribute("Surname"))

規則內邏輯的運作方式

當您要以預設的密碼建立「使用者」物件時，可以使用此規則。在「使用者」物件建立期間，會以「使用者」物件的 Given Name 屬性加上 Surname 屬性為「使用者」物件設定密碼。

您可以透過編輯引數的方式變更預設的密碼值。您可以透過「引數產生器」將密碼設定為任何其他值。

事件轉換：範圍過濾：包括子網路樹

排除指定的子網路樹之外發生的所有事件。您可以在驅動程式內的「訂閱者事件轉換」規則 (Policy) 或「發行者事件轉換」規則 (Policy) 上實作規則 (Rule)。

使用預先定義的規則包含兩個步驟：在「事件轉換」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「事件轉換」規則 (Policy)，請跳至「輸入預先定義的規則」，第 216 頁。

建立規則



- 1 開啟想要管理之驅動程式的「Identity Manager 驅動程式概觀」。
- 2 按一下「發行者」或「訂閱者」通道上的「事件轉換規則」物件。
- 3 按一下「插入」。
- 4 為規則命名，請確定使用「規則產生器」實作規則，然後按一下「確定」。

此時會啟動「規則產生器」。

輸入預先定義的規則

- 1 在「規則產生器」中，按一下「插入」。
- 2 選取「事件轉換：範圍過濾：包括子網路樹」。
- 3 在「規則產生器」中按一下「事件轉換：範圍過濾：包含子網路樹」，以編輯規則。
- 4 刪除「值欄位」內的「/ 輸入要包含的子網路樹 /」。
- 5 按一下瀏覽按鈕，瀏覽想要同步化事件的部份 Identity Vault 網路樹，然後按一下「確定」。
- 6 按一下「確定」。

事件轉換：範圍過濾：包括子網路樹

條件
 if source DN not in subtree "[輸入要包含的子網路樹]"
動作
 veto()

規則內邏輯的運作方式

當您只想要將 Identity vault 與已連接系統之間的特定子網路樹同步化時，可以使用此規則。當某事件未在 Identity Vault 的指定部份發生時，該事件將會被否決。您可以藉由複製及貼上「If Source DN」，第 245 頁 條件的方式新增其他要同步化的子網路樹。

事件轉換：範圍過濾：排除子網路樹

排除發生在指定子網路樹內的所有事件。您可以在驅動程式內的「訂閱者事件轉換」或「發行者事件轉換」規則 (Policy) 上實作規則 (Rule)。

使用預先定義的規則包含兩個步驟：在「事件轉換」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「事件轉換」規則 (Policy)，請跳至「輸入預先定義的規則」，第 217 頁。

建立規則



- 1 開啓想要管理之驅動程式的「Identity Manager 驅動程式概觀」。
- 2 按一下「發行者」或「訂閱者」通道上的「事件轉換規則」物件。
- 3 按一下「插入」。
- 4 為規則命名，請確定使用「規則產生器」實作規則，然後再按一下「確定」。

此時會啓動「規則產生器」。

輸入預先定義的規則

- 1 在「規則產生器」中，按一下「插入」。
- 2 選取「事件轉換：範圍過濾：排除子網路樹」。
- 3 在「規則產生器」中按一下「事件轉換：範圍過濾：排除子網路樹」，以編輯規則。
- 4 排除「值欄位」內的「[/ 輸入要排除的子網路樹/]」。
- 5 按一下瀏覽按鈕，瀏覽想要從同步化排除之事件的部分 Identity Vault 網路樹，然後按一下「確定」。
- 6 按一下「確定」。

事件轉換：範圍過濾：排除子網路樹

條件
 if source DN in subtree "[輸入要排除的子網路樹]"
動作
 veto()

規則內邏輯的運作方式

如果您希望在同步化時，排除部份 Identity Vault 或連接系統，可以使用此規則。在事件發生於 Identity Vault 之該特定部份時，該事件將會被否決。您可以藉由複製及貼上 If Source DN 條件的方式新增其他要排除的子網路樹。

輸入或輸出轉換：將「電話號碼」從 (nnn) nnn-nnnnn 重新格式化為 nnn-xxx-xxxx

轉換電話號碼的格式。在驅動程式內的「輸入或輸出轉換」規則 (Policy) 上實作規則 (Rule)。通常，如果此規則 (Rule) 用在「輸入轉換」上，您可以接著在「輸出轉換」上使用「將「電話號碼」從 nnn-xxx-xxxx 重新格式化為 (nnn) nnn-nnnnn」的規則，反之也可以將格式來回轉換。

使用預先定義的規則包含兩個步驟：在「輸入或輸出轉換」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「輸入或輸出轉換」規則 (Policy)，請跳至「輸入預先定義的規則」，第 218 頁。

建立規則

- 1 開啟想要管理之驅動程式的「Identity Manager 驅動程式概觀」。
- 2 按一下「發行者」或「訂閱者」通道上的「輸入或輸出轉換規則」物件。
- 3 按一下「插入」。
- 4 為規則命名，請確定使用「規則產生器」實作規則，然後再按一下「確定」。

此時會啟動「規則產生器」。

輸入預先定義的規則

- 1 在「規則產生器」中，按一下「插入」。
- 2 選取「輸入或輸出轉換：將電話號碼從 (nnn) nnn-nnnnn 重新格式化為 nnn-xxx-xxxx」。
- 3 在「規則產生器」中，按一下「輸入或輸出轉換：將電話號碼從 (nnn) nnn-nnnnn 重新格式化為 nnn-xxx-xxxx」，以編輯規則。
- 4 定義要執行重新格式化電話號碼的條件。
- 5 按一下「確定」。

[輸入或輸出轉換：將電話號碼從 \(nnn\) nnn-nnnnn 重新格式化為 nnn-xxx-xxxx](#)

條件

🔄 這項條件評估為 true

動作

🔄 reformat operation attribute("phone", Replace First("(\\d\\d\\d)\\s*(\\d\\d\\d)-(\\d\\d\\d\\d)\$", "\$1-\$2-\$3", Local Variable("current-value")))

規則內邏輯的運作方式

當您想要重新格式化電話號碼時，可以使用此規則。它會在目前操作中尋找模式符合 (nnn) nnn-nnnnn 的電話號碼屬性值，並全部取代為 nnn-xxx-xxxx。

輸入或輸出轉換：將電話號碼從 `nnn-nnn-nnnn` 重新格式化為 `(nnn) nnn-nnnn`

轉換電話號碼的格式。在「輸入或輸出轉換」規則 (Policy) 上實作規則 (Rule)。通常，如果您在「輸出轉換」上使用此規則 (Rule)，您可以接著在「輸入轉換」上使用「將電話號碼從 `(nnn) nnn-nnnn` 重新格式化為 `nnn-nnn-nnnn`」的規則，反之也可以將格式來回轉換。

使用預先定義的規則包含兩個步驟：在「輸入或輸出轉換」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「輸入或輸出轉換」規則 (Policy)，請跳至「輸入預先定義的規則」，第 219 頁。

建立規則

- 1 開啟想要管理之驅動程式的「Identity Manager 驅動程式概觀」。
- 2 按一下「發行者」或「訂閱者」通道上的「輸入或輸出轉換規則」物件。
- 3 按一下「插入」。
- 4 為規則命名，請確定使用「規則產生器」實作規則，然後再按一下「確定」。

此時會啟動「規則產生器」。

輸入預先定義的規則

- 1 在「規則產生器」中，按一下「插入」。
- 2 選取「輸入或輸出轉換：將電話號碼從 `nnn-nnn-nnnn` 重新格式化為 `(nnn) nnn-nnnn`」。
- 3 在「規則產生器」中，按一下「輸入或輸出轉換：將電話號碼從 `nnn-nnn-nnnn` 重新格式化為 `(nnn) nnn-nnnn`」，以編輯規則。
- 4 定義要執行重新格式化電話號碼的條件。
- 5 按一下「確定」。

輸入或輸出轉換：將電話號碼從 `nnn-nnn-nnnn` 重新格式化為 `(nnn)`

`nnn-nnnn`

條件

🔄 這項條件評估為 true。

動作

🔄 reformat operation attribute("phone",Replace First("(^(\d\d\d)-(\d\d\d)-(\d\d\d\d)\$)","(\$1) \$2-\$3",Local Variable("current-value")))

規則內邏輯的運作方式

當您想要重新格式化電話號碼時，可以使用此規則。它會在目前操作中尋找模式符合 `(nnn) nnn-nnnn` 的電話號碼屬性值，並全部取代為 `nnn-nnn-nnnn`。

比對：鏡像複製發行者

根據名稱和位置，在 Identity Vault 內尋找已連接系統中的相符物件。在驅動程式內的「發行者相符」規則 (Policy) 上實作規則 (Rule)。

使用預先定義的規則包含兩個步驟：在「相符」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「相符」規則 (Policy)，請跳至「輸入預先定義的規則」，第 220 頁。

建立規則

- 1 開啟想要管理之驅動程式的「*Identity Manager* 驅動程式概觀」。
- 2 按一下「發行者」通道上的「相符規則」物件。
- 3 按一下「插入」。
- 4 為規則命名，請確定使用「規則產生器」實作規則，然後再按一下「確定」。

此時會啟動「規則產生器」。

輸入預先定義的規則

- 1 在「規則產生器」中，按一下「插入」。
- 2 選取「比對：鏡像複製發行者」。
- 3 按一下「規則產生器」內的「比對：鏡像複製發行者」，以編輯規則。
- 4 從「值」欄位中刪除 [輸入來源階層的基礎]。
- 5 在來源階層中瀏覽至要讓相符規則啟動的容器，並加以選取，然後按一下「確定」。
- 6 按一下「確定」。
- 7 從「輸入字串」欄位中刪除 [輸入目的階層的基礎]。
- 8 按一下「編輯引數」圖示以啟動「引數產生器」。
- 9 選取「名詞」清單中選取「文字」，然後再按一下「新增」。
- 10 在編輯器中按一下瀏覽圖示，瀏覽至要與來源結構相符之目的階層內的容器並加以選取，然後按一下「確定」。
- 11 按一下「確定」。

比對：鏡像複製發行者

條件

if source DN in subtree "[輸入來源階層的基礎]"

動作

```
set local variable("dest-base","[輸入目的階層的基礎]")
find matching object(scope="entry",dn(Local Variable("dest-base")+"+Unmatched Source DN
(convert="true")))
```

規則內邏輯的運作方式

當「新增」事件發生在位於指定來源子網路樹中已連接系統的物件上時，此規則會在指定的目的子網路樹 **Identity Vault** 中的相對位置，建構一個代表相同物件名稱的 DN。如果目的物件存在，且是所需的物件類別，就會被視為相符。您必須提供來源（已連接系統）和目的（**Identity Vault**）子網路樹的 DN。

比對：鏡像複製訂閱者：LDAP 格式

根據名稱和位置，在使用 LDAP 格式 DN 的已連接系統中，尋找 **Identity Vault** 中相符的物件。在驅動程式內的「訂閱者相符」規則 (Policy) 上實作規則 (Rule)。

使用預先定義的規則包含兩個步驟：在「相符」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「相符」規則 (Policy)，請跳至「輸入預先定義的規則」，第 221 頁。

建立規則

- 1 開啟想要管理之驅動程式的「*Identity Manager* 驅動程式概觀」。
- 2 按一下「訂閱者」通道上的「相符規則」物件。
- 3 按一下「插入」。
- 4 為規則命名，請確定使用「規則產生器」實作規則，然後再按一下「確定」。

此時會啟動「規則產生器」。

輸入預先定義的規則

- 1 在「規則產生器」中，按一下「插入」。
- 2 選取「比對：鏡像複製訂閱者：LDAP 格式」。
- 3 按一下「規則產生器」內的「比對：鏡像複製訂閱者：LDAP 格式」，以編輯規則。
- 4 從「值」欄位中刪除 [輸入來源階層的基礎]。
- 5 在來源階層中瀏覽至要讓相符規則啟動的容器，並加以選取，然後按一下「確定」。
- 6 按一下「確定」。
- 7 從「輸入字串」欄位中刪除 [輸入目的階層的基礎]。
- 8 按一下「編輯引數」圖示以啟動「引數產生器」。
- 9 選取「名詞」清單中選取「文字」，然後再按一下「新增」。
- 10 在編輯器中按一下瀏覽圖示，瀏覽至要與來源結構相符之目的階層內的容器並加以選取，然後按一下「確定」。
- 11 按一下「確定」。

比對：鏡像複製訂閱者：LDAP 格式

條件

if source DN in subtree "[輸入來源階層的基礎]"

動作

```
set local variable("dest-base","[輸入目的階層的基礎]")
find matching object(scope="entry",dn(Unmatched Source DN(convert="true")+Local Variable("dest-base")))
```

規則內邏輯的運作方式

當「新增」事件發生在位於指定來源子網路樹中 *Identity Vault* 的物件上時，此規則會在指定的目的子網路樹已連接系統中的相對位置，建構一個代表相同物件名稱的 DN。如果目的物件存在，且是所需的物件類別，就會被視為相符。您必須提供來源 (*Identity Vault*) 和目的 (已連接系統) 子網路樹的 DN。已連接系統必須使用 LDAP 格式的 DN。

比對：依屬性值

依指定的屬性值尋找相符的物件。您可以在驅動程式內的「訂閱者相符」規則 (Policy) 或「發行者相符」規則 (Policy) 上實作規則 (Rule)。

使用預先定義的規則包含兩個步驟：在「相符」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「相符」規則 (Policy)，請跳至「輸入預先定義的規則」，第 222 頁。

建立規則

- 1 開啟想要管理之驅動程式的「*Identity Manager* 驅動程式概觀」。
- 2 按一下「發行者」通道上的「相符規則」物件。
- 3 按一下「插入」。
- 4 為規則命名，請確定使用「規則產生器」實作規則，然後再按一下「確定」。

此時會啟動「規則產生器」。

輸入預先定義的規則

- 1 在「規則產生器」中，按一下「插入」。
- 2 選取「比對：依屬性值」。
- 3 按一下「規則產生器」內的「比對：依屬性值」，以編輯規則。
- 4 從「輸入 DN」欄位中刪除 [輸入要開始搜尋的基礎 DN]。
- 5 按一下「編輯引數」圖示以啟動「引數產生器」。
- 6 選取「名詞」清單中選取「文字」，然後按一下「新增」。
- 7 在「編輯器」中按一下瀏覽圖示，然後瀏覽至您希望搜尋啟動的容器並加以選取，然後按一下「確定」。
- 8 從「輸入相符屬性」欄位中刪除「[輸入相符的屬性名稱]」。
- 9 按一下「編輯引數」圖示以啟動「相符屬性產生器」。
- 10 按一下瀏覽圖示並選取希望與其相符的屬性。您可以選取一或多個與其相符的屬性，然後按一下「確定」。
- 11 按一下「確定」。



規則內邏輯的運作方式

當「新增」事件發生在來源資料儲存中的物件上時，此規則會在目的資料儲存中搜尋指定屬性具有相同值的物件。您必須提供要在已連接系統中搜尋的子網路樹基礎 DN，以及要比對的屬性名稱。

佈置：鏡像複製發行者

根據來自已連接系統的名稱和位置，在 Identity Vault 內放置物件。在驅動程式內的「發行者佈置」規則 (Policy) 上實作規則 (Rule)。

使用預先定義的規則包含兩個步驟：在「佈置」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「佈置」規則 (Policy)，請跳至「[輸入預先定義的規則](#)」，第 223 頁。

建立規則

- 1 開啟想要管理之驅動程式的「*Identity Manager* 驅動程式概觀」。
- 2 按一下「發行者」通道上的「佈置規則」物件。
- 3 按一下「插入」。
- 4 為規則命名，請確定使用「規則產生器」實作規則，然後再按一下「確定」。

此時會啟動「規則產生器」。

輸入預先定義的規則

- 1 在「規則產生器」中，按一下「插入」。
- 2 選取「佈置：鏡像複製發行者」。
- 3 按一下「規則產生器」內的「佈置：鏡像複製發行者」，以編輯規則。
- 4 從「值」欄位中刪除 [輸入來源階層的基礎]。
- 5 在來源階層中瀏覽至要讓物件採取動作的容器並加以選取，然後按一下「確定」。
- 6 從「輸入字串」欄位中刪除 [輸入目的階層的基礎]。
- 7 按一下「編輯引數」圖示以啟動「引數產生器」。
- 8 選取「名詞」清單中選取「文字」，然後按一下「新增」。
- 9 在「編輯器」中按一下瀏覽圖示，瀏覽至您希望放置物件之目的階層內的容器並加以選取，然後按一下「確定」。
- 10 按一下「確定」。

佈置：鏡像複製發行者

條件

if source DN in subtree "[輸入來源階層的基礎]"

動作

set local variable("dest-base","[輸入目的階層的基礎]")

set operation destination DN(dn(Local Variable("dest-base")+"+Unmatched Source DN (convert="true")))

規則內邏輯的運作方式

如果「使用者」物件位於指定的已連接系統之來源子網路樹中，則此物件會被放置在 Identity Vault 內相同的相對名稱和位置。您必須提供來源（已連接系統）和目的 (Identity Vault) 子網路樹的 DN。

佈置：鏡像複製訂閱者：LDAP 格式

使用指定點之 Identity Vault 內的鏡像複製結構，將物件放置在資料儲存中。在驅動程式內的「佈置」規則 (Policy) 上實作規則 (Rule)。您只可以在「訂閱者」通道上實作規則。

使用預先定義的規則包含兩個步驟：在「佈置」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「佈置」規則 (Policy)，請跳至「輸入預先定義的規則」，第 224 頁。

建立規則

- 1 開啟想要管理之驅動程式的「*Identity Manager* 驅動程式概觀」。
- 2 按一下「訂閱者」通道上的「佈置規則」物件。
- 3 按一下「插入」。
- 4 為規則命名，請確定使用「規則產生器」實作規則，然後再按一下「確定」。

此時會啟動「規則產生器」。

輸入預先定義的規則

- 1 在「規則產生器」中，按一下「插入」。
- 2 選取「佈置：鏡像複製訂閱者：LDAP 格式」。
- 3 按一下「規則產生器」內的「佈置：鏡像複製訂閱者：LDAP 格式」，以編輯規則。
- 4 從「值」欄位中刪除 [輸入來源階層的基礎]。
- 5 在來源階層中瀏覽至要讓物件採取動作的容器，並加以選取，然後按一下「確定」。
- 6 從「輸入字串」欄位中刪除 [輸入目的階層的基礎]。
- 7 按一下「編輯引數」圖示以啟動「引數產生器」。
- 8 選取「名詞」清單中選取「文字」，然後按一下「新增」。
- 9 在「編輯器」中按一下瀏覽圖示，瀏覽至您希望放置物件之目的階層內的容器並加以選取，然後按一下「確定」。
- 10 按一下「確定」。

佈置：鏡像複製訂閱者：LDAP 格式

條件

if source DN in subtree "[輸入來源階層的基礎]"

動作

set local variable("dest-base","[輸入目的階層的基礎]")

set operation destination DN(dn(Unmatched Source DN(convert="ture")+","+Local variable ("dest-base")))

規則內邏輯的運作方式

如果「使用者」物件位於指定的來源子網路樹中，則此物件會被放置在 Identity Vault 內相同的相對名稱和位置。您必須提供來源 (Identity Vault) 和目的 (已連接系統) 子網路樹的 DN。已連接系統必須使用 LDAP 格式的 DN。

佈置：一般發行者

將來自資料儲存的物件放置到 Identity Vault 內的某個容器中。在驅動程式內的「發行者佈置」規則 (Policy) 上實作規則 (Rule)。

使用預先定義的規則包含兩個步驟：在「佈置」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「佈置」規則 (Policy)，請跳至「輸入預先定義的規則」，第 225 頁。

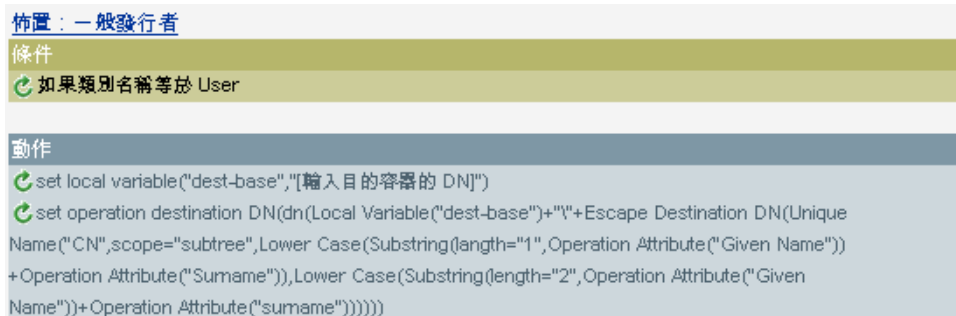
建立規則

- 1 開啟想要管理之驅動程式的「*Identity Manager* 驅動程式概觀」。
- 2 按一下「發行者」通道上的「佈置規則」物件。
- 3 按一下「插入」。
- 4 為規則命名，請確定使用「規則產生器」實作規則，然後再按一下「確定」。

此時會啟動「規則產生器」。

輸入預先定義的規則

- 1 在「規則產生器」中，按一下「插入」。
- 2 選取「佈置：一般發行者」。
- 3 按一下「規則產生器」內的「佈置：一般發行者」，以編輯規則。
- 4 從「輸入字串」欄位中刪除「[輸入目的容器的 DN]」。
- 5 按一下「編輯引數」圖示以啟動「引數產生器」。
- 6 選取「名詞」清單中選取「文字」，然後按一下「新增」。
- 7 在「編輯器」中按一下瀏覽圖示，然後瀏覽至要放置所有使用者物件之目的容器並加以選取，然後按一下「確定」。
- 8 按一下「確定」。



The screenshot shows the configuration interface for a rule. It is titled "佈置：一般發行者" (Configuration: General Issuer). Under the "條件" (Conditions) section, there is a condition: "如果類別名稱等於 User" (If class name equals User). Under the "動作" (Actions) section, there are two actions: "set local variable('dest-base','[輸入目的容器的 DN]')" and "set operation destination DN(dn(Local Variable('dest-base')+\""+Escape Destination DN(Unique Name('CN',scope='subtree',Lower Case(Substring(length='1',Operation Attribute('Given Name'))+Operation Attribute('Surname')),Lower Case(Substring(length='2',Operation Attribute('Given Name'))+Operation Attribute('surname'))))))".

規則內邏輯的運作方式

此規則會將所有「使用者」物件放置到目的 DN 中。此規則會將目的容器的 DN 設定為本地變數 `dest-base`。此規則接著會將目的 DN 設定為 `dest-base\CN` 屬性。「使用者」物件的 CN 屬性值由 `Given Name` 屬性的前兩個字母加上 `Surname` 屬性的小寫字母組成。此規則採用斜線格式。

佈置：一般訂閱者：LDAP 格式

將來自 *Identity Vault* 的物件放置到資料儲存內的某個容器中。在驅動程式內的「訂閱者佈置」規則 (Policy) 上實作規則 (Rule)。

使用預先定義的規則包含兩個步驟：在「佈置」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「佈置」規則 (Policy)，請跳至「輸入預先定義的規則」，第 226 頁。

建立規則

- 1 開啟想要管理之驅動程式的「*Identity Manager* 驅動程式概觀」。

- 2 按一下「訂閱者」通道上的「佈置規則」物件。
- 3 按一下「插入」。
- 4 為規則命名，請確定使用「規則產生器」實作規則，然後再按一下「確定」。

此時會啟動「規則產生器」。

輸入預先定義的規則

- 1 在「規則產生器」中，按一下「插入」。
- 2 選取「佈置：一般訂閱者：LDAP 格式」。
- 3 按一下「規則產生器」內的「佈置：一般訂閱者：LDAP 格式」，以編輯規則。
- 4 從「輸入字串」欄位中刪除「[輸入目的容器的 DN]」。
- 5 按一下「編輯引數」圖示以啟動「引數產生器」。
- 6 選取「名詞」清單中選取「文字」，然後按一下「新增」。
- 7 在編輯器中，新增您希望放置所有「使用者」物件的目的容器。請確定使用 LDAP 格式指定容器，然後按一下「確定」。
- 8 按一下「確定」。



規則內邏輯的運作方式

此規則會將所有「使用者」物件放置到目的 DN 中。此規則會將目的容器的 DN 設定為本地變數 `dest-base`。此規則接著會將目的 DN 設定為 `uid=` 唯一名稱，`dest-base`。「使用者」物件的 `uid` 屬性值由 `Given Name` 屬性的前兩個字母加上 `Surname` 屬性的小寫字母組成。此規則採用 LDAP 格式。

佈置：依部門的發行者

將來自資料儲存內某容器的物件依 OU 屬性值放置到 Identity Vault 的多個容器中。在驅動程式內的「發行者佈置」規則 (Policy) 上實作規則 (Rule)。

使用預先定義的規則包含兩個步驟：在「佈置」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「佈置」規則 (Policy)，請跳至「輸入預先定義的規則」，第 227 頁。

建立規則

- 1 開啓想要管理之驅動程式的「Identity Manager 驅動程式概觀」。
- 2 按一下「發行者」通道上的「佈置規則」物件。
- 3 按一下「插入」。

4 為規則命名，請確定使用「規則產生器」實作規則，然後再按一下「確定」。

此時會啟動「規則產生器」。

輸入預先定義的規則

- 1 在「規則產生器」中，按一下「插入」。
- 2 選取「佈置：依部門的發行者」。
- 3 按一下「佈置：依部門的發行者」，以編輯規則。
- 4 從「輸入字串」欄位中刪除「[輸入目的組織的 DN]」。
- 5 按一下「編輯引數」圖示以啟動「引數產生器」。
- 6 選取「名詞」清單中選取「文字」，然後按一下「新增」。
- 7 在「編輯器」中按一下瀏覽圖示，然後瀏覽至 Identity Vault 內的父容器並加以選取。請確定所有的部門容器皆為此 DN 的子容器，然後按一下「確定」。
- 8 按一下「確定」。

佈置：依部門的發行者

條件

- 如果類別名稱等於 User
- 且 if attribute 'OU' available

動作

```
set local variable("dest-base","[輸入目的組織的 DN]")
set operation destination DN(dn(Local Variable("dest-base")+"\ "+Attribute("OU")+"\ "+Escape
Destination DN(Unique Name("CN",scope=subtree",Lower Case(Substring(length="1",Operation
Attribute("Given Name"))+Operation Attribute("Surname")),Lower Case(Substring
(length="2",Operation Attribute("Given Name"))+Operation Attribute("Surname")))))))
```

規則內邏輯的運作方式

此規則會依據儲存在 OU 屬性內的值，將「使用者」物件放置在適當的部門容器中。若需要放置「使用者」物件，並且其擁有可用的 OU 屬性，則「使用者」物件會被放置在 dest-base\OU 屬性的值\CN 屬性中。

dest-base 為本地變數。DN 必須為相對於部門容器的根路徑。其可為組織或組織單位。儲存於 OU 屬性內的值必須為 dest-base 本地變數的子容器名稱。

OU 屬性的值必須為子容器的名稱。如果 OU 屬性不存在，則不會執行此規則。

「使用者」物件的 CN 屬性值由 Given Name 屬性的前兩個字母加上 Surname 屬性的小寫字母組成。此規則採用斜線格式。

佈置：依部門的訂閱者：**LDAP** 格式

將來自 Identity Vault 內某容器的物件依 OU 屬性放置到資料儲存內的多個容器中。在驅動程式內的「佈置」規則 (Policy) 上實作規則 (Rule)。您只可以在「訂閱者」通道上實作規則。

使用預先定義的規則包含兩個步驟：在「佈置」規則 (Policy) 集中建立規則 (Policy)，並輸入預先定義的規則 (Rule)。若您已擁有想要新增此規則 (Rule) 的「佈置」規則 (Policy)，請跳至「輸入預先定義的規則」，第 228 頁。

建立規則

- 1 開啟想要管理之驅動程式的「*Identity Manager* 驅動程式概觀」。
- 2 按一下「訂閱者」通道上的「佈置規則」物件。
- 3 按一下「插入」。
- 4 為規則命名，請確定使用「規則產生器」實作規則，然後再按一下「確定」。

此時會啟動「規則產生器」。

輸入預先定義的規則

- 1 在「規則產生器」中，按一下「插入」。
- 2 選取「佈置：依部門的訂閱者：LDAP 格式」。
- 3 按一下「規則產生器」內的「佈置：依部門的訂閱者：LDAP 格式」，以編輯規則。
- 4 從「輸入字串」欄位中刪除「[輸入目的組織的 DN]」。
- 5 按一下「編輯引數」圖示以啟動「引數產生器」。
- 6 選取「名詞」清單中選取「文字」，然後按一下「新增」。
- 7 在「編輯器」中新增資料儲存內的父容器。父容器必須以 LDAP 格式指定。請確定所有的部門容器皆為此 DN 的子容器，然後按一下「確定」。
- 8 按一下「確定」。

佈置：依部門的訂閱者：LDAP 格式

條件

- 如果類別名稱等於 User
- 且 if attribute 'OU' available

動作

```
set local variable("dest-base",[輸入目的組織的 DN])
set operation destination DN(dn("uid="+Escape Destination DN(Unique Name
("uid",scope="subtree",Lower Case(Substring(length="1",Operation Attribute("Given Name"))
+Operation Attribute("Surname")),Lower Case(Substring(length="2",Operation Attribute("Given
Name"))+Operation Attribute("Surname")))+",ou="+Attribute("OU")+","+Local Variable(dest-base)))
```

規則內邏輯的運作方式

此規則會依據儲存在 OU 屬性內的值，將「使用者」物件放置在適當的部門容器中。若需要放置「使用者」物件，並且其擁有可用的 OU 屬性，則「使用者」物件會被放置在 uid=唯一名稱,ou=OU 屬性的值,dest-base 中。

dest-base 為本地變數。DN 必須為相對於部門容器的根路徑。其可為組織或組織單位。儲存於 OU 屬性內的值必須為 dest-base 本地變數的子容器名稱。

OU 屬性的值必須為子容器的名稱。如果 OU 屬性不存在，則不會執行此規則。

「使用者」物件的 uid 屬性值由 Given Name 屬性的前兩個字母加上 Surname 屬性的小寫字母組成。此規則採用 LDAP 格式。

3.3 一般運算式

一般運算式是遵循某些模式之相符文字字串的公式。一般運算式是由一般字元和中繼字元所組成。一般字元包含大寫和小寫字母與數字。中繼字元具有特殊意義。下表包含最常見的中繼字元及其代表的意義。

表格 3-1 常見的一般運算式

中繼字元	描述
.	符合任何單一字元。
\$	符合一行的末端。
^	符合一行的起始處。
*	符合緊接著前方之零次或多次出現的字元。
\	常值逸出字元 此字元可讓您搜尋任何中繼字元。例如，\ \$ 尋找 \$1000 ，而不是符合一行的末端。
[]	符合括號間的任一字元。
[0-9]	符合以連字號連接的字元範圍。此範例為符合任一數字。
[A-Za-z]	同時符合多個範圍。此範例為符合所有大寫及小寫的字母。

「引數產生器」專門設計為依照 Java^{*} 內定義的方式使用一般運算式。[Java 網站 \(http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html\)](http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html) 包含進一步的資訊。

3.4 XPath 1.0 運算式

某些條件、動作和記號使用 XPath 1.0 運算式的引數。XPath 是一種專為提供 XSLT 和 Xpointer 之間共用功能之常見語法及語意而建立的語言。其主要用於處理 XML 文件的部份，但也提供處理字串、數字和布林值的基本功能。

XPath 規格需要內嵌式應用程式提供網路位置與數個應用程式定義的訊息片段。在 DirXML 程序檔中 (請參閱「[DirXML 程序檔](#)」，第 11 頁)，XPath 用於評估以下的網路位置：

- ◆ 網路位置節點是目前的操作。
- ◆ 網路位置的位置及大小為 1。
- ◆ 可用變數。
 - ◆ 可做為 Identity Manager 內樣式表參數 (目前有 fromNDS、srcQueryProcessor、destQueryProcessor、srcCommandProcessor、destCommandProcessor 和 dnConverter) 的變數。
 - ◆ 全域組態變數。
 - ◆ 本地規則變數。
 - ◆ 若不同的變數來源之間發生名稱上的衝突，則其優先順序為本地變數、樣式表參數、全域變數。
- ◆ 在規則元素上宣告的名稱空間。

- ◆ 可用的功能
 - ◆ 所有內建的 XPath 1.0 功能
 - ◆ 由 NXSL 提供的 Java 延伸功能
 - ◆ 與具有 Java 類別字首關聯的名稱空間宣告必須在規則元素上宣告。

W3 網站 (<http://www.w3.org/TR/1999/REC-xpath-19991116>) 包含進一步的資訊。

3.5 條件

本節包含使用「規則產生器」介面時，可用的所有條件之詳細參考資訊。

- ◆ 「If 關聯」，第 230 頁
- ◆ 「If Attribute」，第 231 頁
- ◆ 「If 類別名稱」，第 232 頁
- ◆ 「If Destination Attribute」，第 233 頁
- ◆ 「If 目的 DN」，第 234 頁
- ◆ 「If Entitlement」，第 235 頁
- ◆ 「If Global Configuration Value」，第 236 頁
- ◆ 「If Local Variable」，第 237 頁
- ◆ 「If Named Password」，第 239 頁
- ◆ 「If 操作」，第 239 頁
- ◆ 「If Operation Attribute」，第 241 頁
- ◆ 「If 操作內容」，第 242 頁
- ◆ 「If Password」，第 243 頁
- ◆ 「If Source Attribute」，第 243 頁
- ◆ 「If Source DN」，第 245 頁
- ◆ 「If XPath Expression」，第 246 頁

3.5.1 If 關聯

對目前操作或目前物件的關聯值進行測試。

欄位

符合運算子條件的情況

運算子	符合條件的情況
associated	已為目前物件建立關聯。
available	目前操作指定了一個非空白關聯值。
equal	目前操作指定的關聯值完全等於 if 關聯的內容。
not-associated	尚未為目前物件建立關聯。
not available	關聯無法供目前物件使用。

運算子	符合條件的情況
不等於	目前操作指定的關聯值不等於 if 關聯的內容。

範例

此例將測試關聯是否可用。當此條件符合時，即執行定義的動作。



3.5.2 If Attribute

對目前操作或來源資料儲存內目前物件的屬性值進行測試。此屬性在邏輯上可被視為「If 操作屬性」或「If 來源屬性」，因為若符合來源資料儲存或操作內的條件，則測試結果是令人滿意的。

欄位

名稱

指定要測試的屬性名稱。

運算子

選取條件測試類型。

比較模式

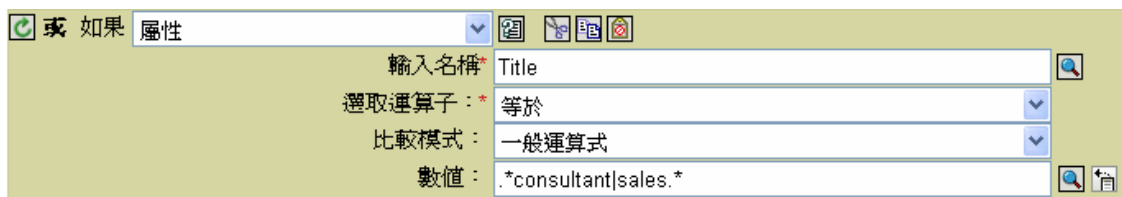
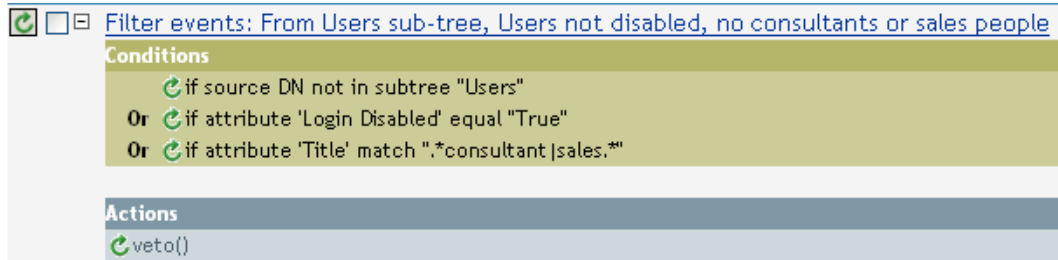
選取比較模式。請參閱「[比較模式](#)」，第 305 頁。

符合運算子條件的情況

運算子	符合條件的情況
available	指定屬性的目前操作或來源資料儲存中內含可用值。
equal	指定屬性的目前操作或來源資料儲存中內含可用值，當使用指定比較模式進行比較時，其等於指定值。
not available	available 將傳回 False。
not-equal	equal 將傳回 False。

範例

此範例在過濾已停用或具有某些職稱的「使用者」物件時，使用了 If Attribute 條件。該項規則為「過濾事件的規則」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。



此條件會尋找任何指定為顧問或銷售人員之具有 Title 屬性的「使用者」物件。

3.5.3 If 類別名稱

對目前操作的物件類別名稱進行測試。

欄位

運算子

選擇條件測試類型。

比較模式

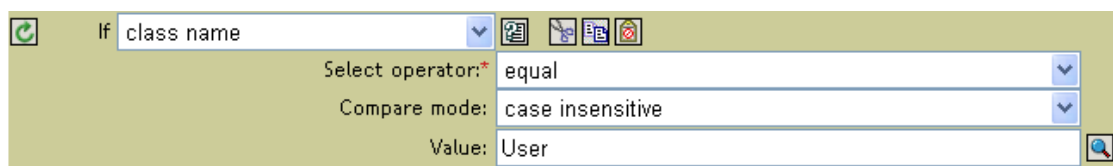
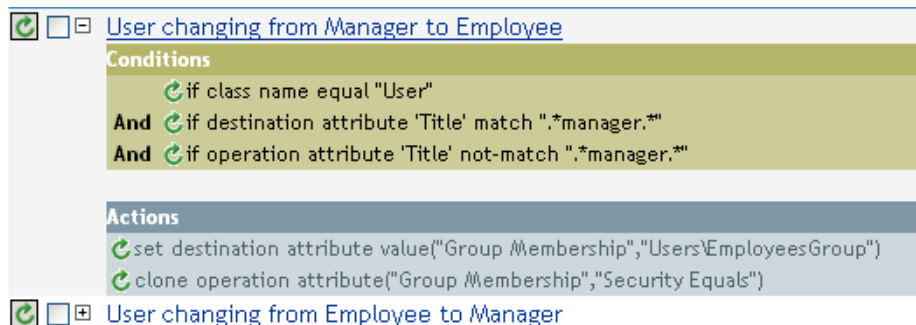
選擇比較模式。請參閱「[比較模式](#)」，第 305 頁。

符合運算子條件的情況

運算子	符合條件的情況
available	目前操作中內含可用的物件類別名稱。
equal	目前操作中內含可用的物件類別名稱，並且在使用指定比較模式進行比較時，其等於指定值。
not available	available 將傳回 False。
not-equal	equal 將傳回 False。

範例

此範例使用 If Class Name 條件來為「使用者」物件依其職稱決定群組成員。該項規則為「依 Title 屬性管理使用者的群組」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「可下載的 Identity Manager 規則」，第 32 頁。



檢查目前操作的類別名稱是否為 User。

3.5.4 If Destination Attribute

對目前操作或目的資料儲存內目前物件的屬性值進行測試。

欄位

名稱

指定要測試的屬性名稱。

運算子

選取條件測試類型。

比較模式

選取比較模式。請參閱「比較模式」，第 305 頁。

符合運算子條件的情況

運算子	符合條件的情況
available	指定屬性的目的資料儲存中內含可用值。
equal	目的資料儲存內之指定屬性中內含可用值，當使用指定比較模式進行比較時，其等於指定值。
not available	available 將傳回 False。
not-equal	equal 將傳回 False。

範例

此範例使用 If Attribute 條件來為「使用者」物件依其職稱決定群組成員。該項規則為「依 Title 屬性管理使用者的群組」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「可下載的 Identity Manager 規則」，第 32 頁。

The screenshot displays two rule configurations in a list. The first rule, 'User changing from Manager to Employee', is expanded to show its conditions and actions. The conditions are: 'if class name equal "User"', 'And if destination attribute 'Title' match ".*manager.*"', and 'And if operation attribute 'Title' not-match ".*manager.*"'. The actions are: 'set destination attribute value("Group Membership","Users\EmployeesGroup")' and 'clone operation attribute("Group Membership","Security Equals")'. Below this, a configuration window for the 'And if' condition is shown, with '目的屬性' selected, 'Title' entered in the '輸入屬性名稱' field, '等於' selected for the '選擇運算子', '一般運算式' selected for the '比較模式', and '.*manager.*' entered in the '數值' field.

此規則會檢查 Title 屬性值是否包含 manager。

3.5.5 If 目的 DN

對目前操作的目的 DN 進行測試。測試會根據指定的運算子執行。

欄位

運算子

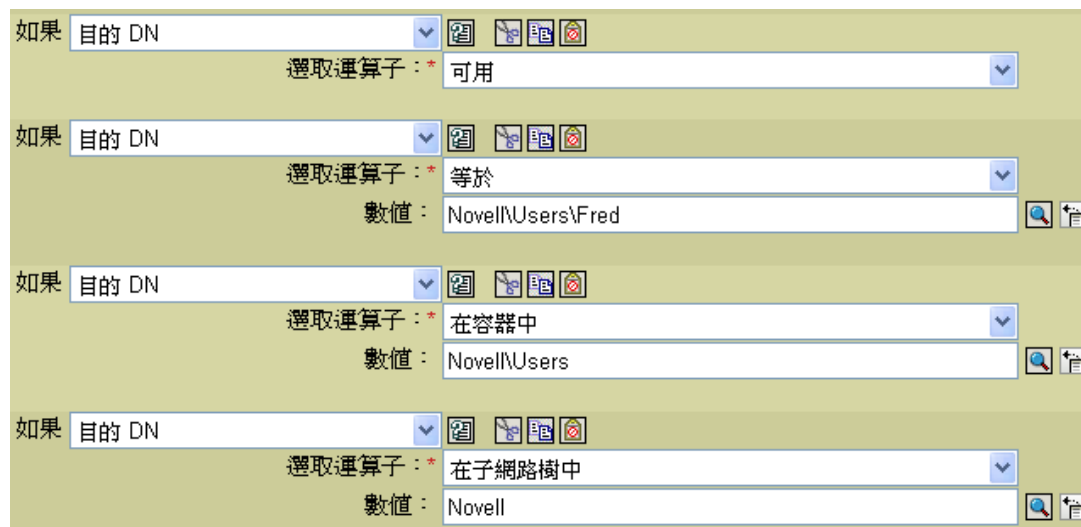
選擇條件測試類型。

符合運算子條件的情況

運算子	符合條件的情況
available	其中內含可用的目的 DN。
equal	其中內含可用的目的 DN，並且當使用適用於目的資料儲存之 DN 格式的語意進行比較時，其等於指定值。
in-container	其中內含可用的目的 DN，並且當使用適用於目的資料儲存之 DN 格式的語意進行比較時，其代表容器內的某個物件，由值指定。
in-subtree	其中內含可用的目的 DN，並且當使用適用於目的資料儲存之 DN 格式的語意進行比較時，其代表子網路樹內的某個物件，由值指定。
not available	available 將傳回 False。
not-equal	equal 將傳回 False。
not-in-container	in-container 將傳回 False。

運算子	符合條件的情況
not-in-subtree	in-subtree 將傳回 False。

範例



3.5.6 If Entitlement

對目前操作或 Identity Vault 內目前物件的授權進行測試。

欄位

名稱

指定要為選取條件進行測試的授權名稱。

運算子

選取條件測試類型。

比較模式

選取比較模式。請參閱「[比較模式](#)」，第 305 頁。

符合運算子條件的情況

運算子	符合條件的情況
available	目前操作或 Identity Vault 內含可用的命名授權。
changing	目前操作包含命名授權的變更 (修改屬性或新增屬性)。
changing-from	目前操作包含命名授權的移除值變更 (移除值)，當使用指定比較模式進行比較時，其中內含的值等於指定值。
changing-to	目前操作包含對命名授權的新增值 (新增值或新增屬性) 變更。在使用指定比較模式進行比較時，其具有與指定值相等的值。

運算子	符合條件的情況
equal	目的資料儲存內之指定屬性中內含可用值，當使用指定比較模式進行比較時，其等於指定值。
not available	available 將傳回 False。
not-changing	changing 將傳回 False。
not-changing-from	changing-from 將傳回 False。
not-changing-to	changing-to 將傳回 False。
not-equal	equal 將傳回 False。

範例

如果 授權

輸入名稱* notes-group

選取運算子:* 可用

如果 授權

輸入名稱* notes-group

選取運算子:* 變更

如果 授權

輸入名稱* notes-group

選取運算子:* 變更自

比較模式: 不區分大小寫

數值: Sales

如果 授權

輸入名稱* notes-group

選取運算子:* 變更為

比較模式: 不區分大小寫

數值: Sales

如果 授權

輸入名稱* notes-group

選取運算子:* 等於

比較模式: 不區分大小寫

數值: Sales

3.5.7 If Global Configuration Value

對全域組態變數進行測試。

欄位

名稱

指定要為選取條件進行測試的全域變數名稱。

運算子

選取條件測試類型。

比較模式

選取比較模式。請參閱「[比較模式](#)」，第 305 頁。

符合運算子條件的情況

運算子	符合條件的情況
available	其中內含使用指定名稱的全域組態變數。
equal	其中內含使用指定名稱的全域組態變數，並且當使用指定比較模式進行比較時，其等於指定值。
not available	available 將傳回 False。
not-equal	equal 將傳回 False。

範例

如果 全域組態值

輸入名稱* myGlobalVariable

選取運算子:* 可用

如果 全域組態值

輸入名稱* myGlobalVariable

選取運算子:* 等於

比較模式: 不區分大小寫

數值: enabled

3.5.8 If Local Variable

對本地變數進行測試。

欄位

名稱

指定要為選取條件進行測試的本地變數名稱。

運算子

選取條件測試類型。

比較模式

選取比較模式。請參閱「[比較模式](#)」，第 305 頁。

符合運算子條件的情況

運算子	符合條件的情況
available	其中內含本地變數，其指定名稱由規則 (Policy) 內舊版規則 (Rule) 的動作加以定義。

運算子	符合條件的情況
equal	其中內含使用指定名稱的本地變數，並且當使用指定比較模式進行比較時，其等於指定值。
not available	available 將傳回 False。
not-equal	equal 將傳回 False。

範例

此範例會依據 Title，將「使用者」物件新增至適當的群組 (Employee 或 Manager)。同時也會視需要建立群組，並設定與該群組相等的安全性。該項規則為「依 Title 屬性管理使用者的群組」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

[Set local variables to test existence of groups and for placement](#)

[Create ManagersGroup, if needed](#)

條件

- 如果 本地變數 'manager-group-info' 可用
- 且** 如果 本地變數 'manager-group-info' 不等於 "group"

動作

- 新增目的物件(類別名稱="group",when="before",dn(本地變數("manager-group-dn")))

[Create EmployeesGroup, if needed](#)

[If Title indicates Manager, add to ManagerGroup and set rights](#)

[If Title does not indicate Manager, add to EmployeeGroup and set rights](#)

此規則 (Policy) 包含五個互相倚賴的規則 (Rule)。

[Set local variables to test existence of groups and for placement](#)

條件

- 如果 類別名稱 等於 "User"
- 且**
- 如果 操作 等於 "add"
- 或** 如果 操作 等於 "modify"

動作

- 設定本地變數("manager-group-dn","Users\ManagersGroup")
- 設定本地變數("manager-group-info",目的屬性("Object Class",dn(本地變數("manager-group-dn"))))
- 設定本地變數("employee-group-dn","Users\EmployeesGroup")
- 設定本地變數("employee-group-info",目的屬性("Object Class",dn(本地變數("employee-group-dn"))))

爲了要讓 If Locate Variable 條件運作，第一個規則會設定四個用以測試群組的不同本地變數，以及放置群組的地方。

規則尋找的條件是要檢查 manager-group-info 的本地變數是否可用，以及 manager-group-info 是否不等於群組。若這些條件符合，則會新增群組的目的物件。

3.5.9 If Named Password

對使用指定名稱之目前操作的密碼進行測試。

欄位

名稱

指定要爲選取條件進行測試的具名密碼名稱。

運算子

選取條件測試類型。

符合運算子條件的情況

運算子	符合條件的情況
available	其中內含使用可用之指定名稱的密碼。
not available	available 將傳回 False。

範例

3.5.10 If 操作

對目前操作的名稱進行測試。

欄位

運算子

選取條件測試類型。

符合運算子條件的情況

運算子	符合條件的情況
equal	目前操作的名稱完全等於 If Operation 的內容。
not-equal	equal 將傳回 False。

值

其值為 Metadirectory 引擎在此條件中尋找的操作：

- ◆ add
- ◆ add-association
- ◆ check-object-password
- ◆ delete
- ◆ get-named-password
- ◆ modify
- ◆ modify-association
- ◆ modify-password
- ◆ move
- ◆ init-params
- ◆ instance

範例

此範例會依據 Title，將「使用者」物件新增至適當的群組 (Employee 或 Manager)。同時也會視需要建立群組，並設定與該群組相等的安全性。該項規則名稱爲「依 Title 屬性管理使用者的群組」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

The screenshot shows a rule configuration window titled "Set local variables to test existence of groups and for placement". It is divided into two main sections: "條件" (Conditions) and "動作" (Actions).

條件 (Conditions):

- 如果 類別名稱 等於 "User"
- 且
- 如果 操作 等於 "add"
- 或 如果 操作 等於 "modify"

動作 (Actions):

- 設定本地變數("manager-group-dn", "Users\ManagersGroup")
- 設定本地變數("manager-group-info", 目的屬性("Object Class", dn(本地變數("manager-group-dn"))))
- 設定本地變數("employee-group-dn", "Users\EmployeesGroup")
- 設定本地變數("employee-group-info", 目的屬性("Object Class", dn(本地變數("employee-group-dn"))))

At the bottom, there is a configuration panel for the "如果 操作" (If Operation) condition. It shows a dropdown menu for "選擇運算子" (Select Operator) set to "等於" (equals), and a text field for "數值" (Value) set to "add".

此條件會檢查是否已發生新增或修改操作。當發生這些事件之一時，它會設定本地變數。

3.5.11 If Operation Attribute

對目前操作的屬性值進行測試。測試會根據指定的運算子執行。

欄位

名稱

指定要測試的屬性名稱。

運算子

選取條件測試類型。

比較模式

選取比較模式。請參閱「[比較模式](#)」，第 305 頁。

符合運算子條件的情況

運算子	符合條件的情況
available	指定屬性的目前操作 (新增屬性、新增值、屬性) 中內含可用值。
changing	目前操作包含指定屬性的變更 (修改屬性或新增屬性)。
changing-from	目前操作包含指定屬性的移除值 (移除值) 變更。在使用指定比較模式進行比較時，其等於指定值。
changing-to	目前操作包含對指定屬性的新增值 (新增值或新增屬性) 變更。在使用指定比較模式進行比較時，其等於指定值。
equal	指定屬性的目前操作 (除了移除值之外) 中內含可用值。在使用指定比較模式進行比較時，其等於指定值。
not available	available 將傳回 False。
not-changing	changing 將傳回 False。
not-changing-from	changing-from 將傳回 False。
not-changing-to	changing-to 將傳回 False。
not-equal	equal 將傳回 False。

範例

此範例會依據 Title，將「使用者」物件新增至適當的群組 (Employee 或 Manager)。同時也會視需要建立群組，並設定與該群組相等的安全性。該項規則名稱為「依 Title 屬性管理使

用者的群組」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

The screenshot shows a list of actions and conditions for a rule. The actions include:

- Set local variables to test existence of groups and for placement
- Create ManagersGroup, if needed
- Create EmployeesGroup, if needed
- If Title indicates Manager, add to ManagerGroup and set rights
- If Title does not indicate Manager, add to EmployeeGroup and set rights

The conditions for the 'If Title indicates Manager...' action are:

- if class name equal "User"
- And if operation attribute 'Title' match ".*manager.*"

The screenshot shows a configuration dialog box for an 'If' condition. The dialog has the following fields:

- Input Name: Title
- Select Operator: 等於 (Equal)
- Comparison Mode: 一般運算式 (General Expression)
- Value: .*manager.*

此條件會檢查 Title 屬性是否等於一般運算式 `.*manager.*`。這表示此條件會尋找在經理之前含零個或多個字元，以及經理之後含單一字元的職稱。若「使用者」物件的職稱為銷售經理，便符合此條件。

3.5.12 If 操作內容

對目前操作的操作內容進行測試。

欄位

名稱

指定要為選取條件進行測試的操作內容名稱。

運算子

選取條件測試類型。

比較模式

選取比較模式。請參閱「[比較模式](#)」，第 305 頁。

符合運算子條件的情況

運算子	符合條件的情況
available	目前操作上內含使用指定名稱的操作內容。
equal	目前操作上內含使用指定名稱的操作內容，並且在使用指定比較模式進行比較時，其值等於所提供的內容。
not available	available 將傳回 False。

運算子	符合條件的情況
not-equal	equal 將傳回 False。

範例

3.5.13 If Password

對目前操作的密碼進行測試。

欄位

運算子

選取條件測試類型。

符合運算子條件的情況

運算子	符合條件的情況
available	目前操作中內含可用的密碼。
not available	available 將傳回 False。

範例

3.5.14 If Source Attribute

對目前操作或來源資料儲存內目前物件的屬性值進行測試。

欄位

名稱

指定要為選取條件進行測試的來源屬性名稱。

運算子

選取條件測試類型。

比較模式

選取比較模式。請參閱「[比較模式](#)」，第 305 頁。

符合運算子條件的情況

運算子	符合條件的情況
available	指定屬性的來源資料儲存中內含可用值。
equal	指定屬性的來源資料儲存中內含可用值。在使用指定比較模式進行比較時，其等於指定值。
not available	available 將傳回 False。
not-equal	equal 將傳回 False。

欄位

名稱

指定要為選取條件進行測試的來源屬性名稱。

運算子

選取條件測試類型。

比較模式

選取比較模式。請參閱「[比較模式](#)」，第 305 頁。

符合運算子條件的情況

運算子	符合條件的情況
available	指定屬性的來源資料儲存中內含可用值。
equal	指定屬性的來源資料儲存中內含可用值。在使用指定比較模式進行比較時，其等於指定值。
not available	available 將傳回 False。
not-equal	equal 將傳回 False。

範例

The screenshot shows three identical 'If' condition rows. Each row has a dropdown menu on the left set to '來源屬性'. The first row has '輸入屬性名稱' set to 'OU' and '選取運算子' set to '可用'. The second row has '輸入屬性名稱' set to 'OU', '選取運算子' set to '等於', '比較模式' set to '不區分大小寫', and '數值' set to 'Sales'. The third row has '輸入屬性名稱' set to 'Language', '選取運算子' set to '等於', '比較模式' set to '結構化', and a list box for '結構化的元件' containing 'string(EN)' and 'string(JP)'.

3.5.15 If Source DN

對目前操作的來源 DN 進行測試。

欄位

運算子

選取條件測試類型。

符合運算子條件的情況

運算子	符合條件的情況
available	其中內含可用的來源 DN。
equal	其中內含可用的來源 DN，並且其等於指定值 in-container 的內容。其中內含可用的來源 DN，並且在指定值所識別的容器內，其代表一個物件。
in-subtree	其中內含可用的來源 DN，並且其代表由指定值所識別之子網路樹內的物件。
not available	available 將傳回 False。
not-equal	equal 將傳回 False。
not-in-container	in-container 將傳回 False。
not-in-subtree	in-subtree 將傳回 False。

欄位

運算子

選取條件測試類型。

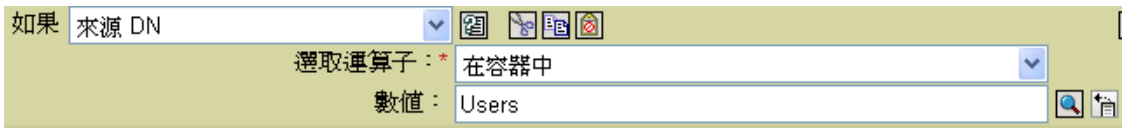
符合運算子條件的情況

運算子	符合條件的情況
available	其中內含可用的來源 DN。
equal	其中內含可用的來源 DN，並且其等於指定值 <code>in-container</code> 的內容。其中內含可用的來源 DN，並且在指定值所識別的容器內，其代表一個物件。
in-subtree	其中內含可用的來源 DN，並且其代表由指定值所識別之子網路樹內的物件。
not available	<code>available</code> 將傳回 <code>False</code> 。
not-equal	<code>equal</code> 將傳回 <code>False</code> 。
not-in-container	<code>in-container</code> 將傳回 <code>False</code> 。
not-in-subtree	<code>in-subtree</code> 將傳回 <code>False</code> 。

範例

此範例使用 If Source DN 條件來檢查位於來源 DN 內的「使用者」物件。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[事件轉換：範圍過濾：排除子網路樹](#)」，第 217 頁。

事件轉換：範圍過濾：排除子網路樹



此條件用於檢查 Users 容器內是否存在來源 DN。若物件來自該容器，則會被否決。

3.5.16 If XPath Expression

對 XPath 1.0 運算式的計算結果進行測試。

欄位

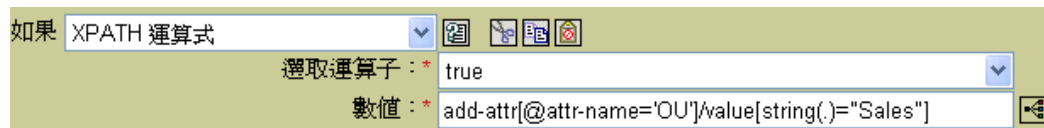
運算子

選取條件測試類型。

符合運算子條件的情況

運算子	符合條件的情況
true	XPath 運算式會評估為 True。
false	true 將傳回 False。

範例



3.6 動作

本節包含使用「規則產生器」介面時，可用的所有動作之詳細參考資訊。

- ◆ 「新增關聯」，第 248 頁
- ◆ 「新增目的屬性值」，第 249 頁
- ◆ 「新增目的物件」，第 250 頁
- ◆ 「新增來源屬性值」，第 251 頁
- ◆ 「新增來源物件」，第 252 頁
- ◆ 「附加 XML 元素」，第 253 頁
- ◆ 「附加 XML 文字」，第 253 頁
- ◆ 「中斷」，第 254 頁
- ◆ 「清除目的屬性值」，第 254 頁
- ◆ 「清除操作內容」，第 255 頁
- ◆ 「清除單次登入 (SSO) 身分證明」，第 255 頁
- ◆ 「清除來源屬性值」，第 256 頁
- ◆ 「由 XPath 運算式複製」，第 256 頁
- ◆ 「複製操作屬性」，第 257 頁
- ◆ 「刪除目的物件」，第 257 頁
- ◆ 「刪除來源物件」，第 258 頁
- ◆ 「尋找相符物件」，第 258 頁
- ◆ 「針對每個」，第 259 頁
- ◆ 「產生事件」，第 260 頁
- ◆ 「實作授權」，第 262 頁
- ◆ 「移動目的物件」，第 263 頁
- ◆ 「移動來源物件」，第 264 頁
- ◆ 「重新格式化操作屬性」，第 265 頁
- ◆ 「移除關聯」，第 265 頁
- ◆ 「移除目的屬性值」，第 266 頁

- ◆ 「移除來源屬性值」，第 267 頁
- ◆ 「重新命名目的物件」，第 267 頁
- ◆ 「重新命名操作屬性」，第 268 頁
- ◆ 「重新命名來源物件」，第 268 頁
- ◆ 「傳送電子郵件」，第 269 頁
- ◆ 「從範本傳送電子郵件」，第 270 頁
- ◆ 「設定預設的屬性值」，第 271 頁
- ◆ 「設定目的屬性值」，第 272 頁
- ◆ 「設定目的密碼」，第 273 頁
- ◆ 「設定本地變數」，第 274 頁
- ◆ 「設定操作關聯」，第 275 頁
- ◆ 「設定操作類別名稱」，第 275 頁
- ◆ 「設定操作目的 DN」，第 275 頁
- ◆ 「設定操作內容」，第 276 頁
- ◆ 「設定操作來源 DN」，第 276 頁
- ◆ 「設定操作範本 DN」，第 277 頁
- ◆ 「設定來源屬性值」，第 277 頁
- ◆ 「設定來源密碼」，第 278 頁
- ◆ 「設定 SSO 身分證明」，第 279 頁
- ◆ 「設定 SSO 通關密語」，第 279 頁
- ◆ 「設定 XML 屬性」，第 280 頁
- ◆ 「設定 SSO 身分證明」，第 281 頁
- ◆ 「狀態」，第 282 頁
- ◆ 「去除操作屬性」，第 282 頁
- ◆ 「去除 XPath」，第 283 頁
- ◆ 「追蹤訊息」，第 283 頁
- ◆ 「否決」，第 284 頁
- ◆ 「若無法使用操作屬性則否決」，第 284 頁

3.6.1 新增關聯

Fields

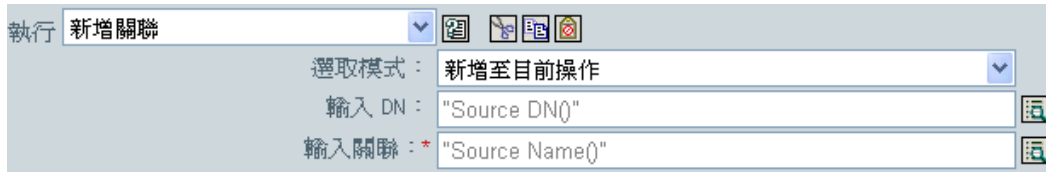
Mode

DN

Association

指定要新增之關聯的值。

範例



3.6.2 新增目的屬性值

為目的資料儲存中物件上的屬性新增值。

欄位

屬性名稱

指定屬性的名稱。

類別名稱

(選擇性) 指定目標物件的類別名稱。保留為空白可使用目前物件的類別名稱。

模式

選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。

物件

選取目標物件。此物件可以是目前物件，也可以由 DN 或關聯指定。

值類型

選取要新增之屬性值的語法。

值

指定要新增的屬性值。

範例

此範例會將目的屬性值新增至 OU 屬性。它會從已建立的本地變數建立其值。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[指令轉換：建立部門容器：第 1 部份和第 2 部份](#)」，第 210 頁。

```
Command Transformation - Create Departmental Container - Part 1
條件
如果操作等於 add
動作
set local variable("target-container",Destination DN(length="-2"))
set local variable("does-target-exist",Destination Attribute("objectclass",class
name="OrganizationalUnit",dn(Local Variable("target-container"))))
```

Command Transformation - Create Departmental Container - Part2

條件

- 如果本地變數 does-target-exist 可供使用
- 且 如果本地變數 does-target-exist 等於 ""

動作

- add destination object(class name="organizationalUnit",direct="true",dn(Local Variable("target-container")))
- add destination attribute value("ou",direct="true",dn(Local Variable("target-container")),Parse DN("dest-dn","dot",length="1",Local Variable("target-container")))

執行	新增目的屬性值	   
輸入屬性名稱 *	ou	
輸入類別名稱		
選取模式	直接寫入至目的資料儲存	
選取物件	DN	
輸入 DN *	本地變數("target-container")	
輸入值類型	string	
輸入字串 *	剖析 DN("dest-dn","dot",長度="1",開始="-1",本地變數("1"))	

3.6.3 新增目的物件

在目的資料儲存內建立指定類型的新物件。

欄位

類別名稱

指定要建立之物件的類別名稱。

模式

選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。

DN

指定要建立之物件的 DN。

備註

在建立物件過程中要新增的任何屬性值，都必須使用相同 DN 在後續的「新增目的屬性值」，第 249 頁動作中完成。

範例

此範例會建立所需的部門容器。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱預先定義規則的「[指令轉換：建立部門容器：第 1 部份和第 2 部份](#)」，[第 210 頁](#)。

The screenshot displays the configuration for a rule named "Command Transformation - Create Departmental Container". It is divided into two parts:

- Part 1:**
 - 條件 (Conditions):** 如果操作等於 add
 - 動作 (Actions):**
 - set local variable("target-container", Destination DN(length="-2"))
 - set local variable("does-target-exist", Destination Attribute("objectclass", class name="OrganizationalUnit", dn(Local Variable("target-container"))))
- Part 2:**
 - 條件 (Conditions):**
 - 如果本地變數 does-target-exist 可供使用
 - 且 如果本地變數 does-target-exist 等於 ""
 - 動作 (Actions):**
 - add destination object(class name="organizationalUnit", direct="true", dn(Local Variable("target-container")))
 - add destination attribute value("ou", direct="true", dn(Local Variable("target-container")), Parse DN("dest-dn", "dot", length="1", Local Variable("target-container")))

Below the rule configuration is a UI control for "新增目的物件" (Add Target Object). It includes the following fields:

- 執行 (Execute) button
- 輸入類別名稱 (*): organizationalUnit
- 選取模式 (Select Mode): 直接寫入至目的資料儲存 (Direct Write to Target Storage)
- 輸入 DN (*): 本地變數("target-container")

已建立 OU 物件。已從此動作後發生之目的屬性值動作建立了 OU 屬性的值。

3.6.4 新增來源屬性值

為來源資料儲存中物件上的指定屬性，加入指定的值。目標物件是目前物件、DN 或關聯。

欄位

屬性名稱

指定屬性的名稱。

類別名稱

(選擇性) 指定目標物件的類別名稱。保留為空白可使用目前物件的類別名稱。

物件

選取目標物件。此物件可以是目前物件，也可以由 DN 或關聯指定。

值類型

選取要新增之屬性值的語法。

值

指定要新增的屬性值。

範例

執行 新增來源屬性值

輸入屬性名稱：* Member

輸入類別名稱：

選取物件：DN

輸入 DN：* "Users/ManagerGroup"

輸入值類型：string

輸入字串：* "Destination DN()"

3.6.5 新增來源物件

建立要建立在來源資料儲存內的指定類型物件。在建立物件過程中要新增的任何屬性值，都必須使用相同 DN 在後續的[新增來源屬性值 \(第 251 頁\)](#) 動作中完成。

欄位

類別名稱

指定要新增之物件的類別名稱。

DN

指定要新增之物件的 DN。

範例

執行 新增來源物件

輸入類別名稱：* User

輸入 DN：* "Users/Fred Flintstone"

執行 新增來源屬性值

輸入屬性名稱：* Surname

輸入類別名稱：

選取物件：DN

輸入 DN：* "Users/Fred Flintstone"

輸入值類型：

輸入字串：* "Flintstone"

欄位

類別名稱

指定要新增至來源資料儲存的物件類別名稱。

DN

指定要新增至來源資料儲存的新物件之 DN。

3.6.6 附加 XML 元素

附加元素到 XPath 運算式選取的元素集中。

欄位

名稱

指定 XML 元素的標記名稱。若先前已在此規則中定義字首，則此名稱可以包含名稱空間字首。

XPATH 運算式

指定傳回節點集的 XPath 1.0 運算式，此節點集包含應附加新元素的元素。

範例

附加 XML 元素	<input type="text" value="jdbc:statement"/>
輸入 XPath 運算式 : *	<input type="text" value=".."/>
附加 XML 元素	<input type="text" value="jdbc:sql"/>
輸入 XPath 運算式 : *	<input type="text" value="../jdbc:statement[last()]"/>
附加 XML 文字	<input +操作內容("a)"="" type="text" value="UPDATE dirxml.emp SET fname = '"/>
輸入 XPath 運算式 : *	<input type="text" value="../jdbc:statement[last()]/jdbc:sql"/>
輸入字串 : *	<input +操作內容("a)"="" type="text" value="UPDATE dirxml.emp SET fname = '"/>
附加 XML 文字	<input +操作內容("a)"="" type="text" value="UPDATE dirxml.emp SET fname = '"/>
輸入 XPath 運算式 : *	<input type="text" value="../jdbc:statement[last()]/jdbc:sql"/>
輸入字串 : *	<input +操作內容("a)"="" type="text" value="UPDATE dirxml.emp SET fname = '"/>

3.6.7 附加 XML 文字

附加文字到 XPath 運算式選取的元素集中。

欄位

XPATH 運算式

指定傳回節點集的 XPath 1.0 運算式，此節點集包含應附加新元素的元素。

字串

指定要附加的文字。

範例

附加 XML 元素	輸入名稱*	jdbc:statement
	輸入 XPATH 運算式 : *	..
附加 XML 元素	輸入名稱*	jdbc:sql
	輸入 XPATH 運算式 : *	../jdbc:statement[last()]
附加 XML 文字	輸入 XPATH 運算式 : *	../jdbc:statement[last()]/jdbc:sql
	輸入字串 :	"UPDATE dirxml.emp SET fname = ""+操作內容("a")"
附加 XML 文字	輸入 XPATH 運算式 : *	../jdbc:statement[last()]/jdbc:sql
	輸入字串 :	"UPDATE dirxml.emp SET fname = ""+操作內容("a")"

3.6.8 中斷

目前規則結束目前操作的處理。

範例

執行	中斷
----	----

3.6.9 清除目的屬性值

從目的資料儲存中的物件移除具名屬性的所有值。

欄位

屬性名稱

指定屬性的名稱。

類別名稱

(選擇性) 指定目標物件的類別名稱。保留為空白可使用目前物件的類別名稱。

模式

選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。

物件

選取目標物件。此物件可以是目前物件，也可以由 DN 或關聯指定。

範例

執行 清除來源屬性值

輸入屬性名稱：* Member

輸入類別名稱：

選取物件：DN

輸入 DN：* "Users\ManagerGroup"

3.6.10 清除操作內容

清除目前操作的任何操作內容。

欄位

內容名稱

指定要清除之操作內容的名稱。

範例

執行 清除操作內容

輸入內容名稱：* myStoredProperty

3.6.11 清除單次登入 (SSO) 身分證明

清除「單次登入」身分證明，以便可以取消提供物件。此動作是「身分證明提供」規則的一部份。如需相關資訊，請參閱第 4 章「Novell 身分證明提供規則」，第 307 頁。

欄位

身分證明儲存物件 **DN**

指定儲存機制物件的 DN。

目標使用者 **DN**

指定目標使用者的 DN。

應用程式身分證明 **ID**

指定儲存在應用程式物件內的應用程式身分證明。

登入參數字串

指定應用程式的每一個登入參數。登入參數是儲存在應用程式物件內的驗證金鑰。

範例

Do clear SSO credential

Enter credential store object DN:* ..\GroupWise_Repository

Render browsed DN relative to policy

Enter target user DN:* Destination Attribute("DirXML-ADContext",class name="U:

[Populate the following from an application object](#)

Enter application credential ID:* GroupWise_Credential

Enter login parameter strings: Username,Password

3.6.12 清除來源屬性值

從來源資料儲存中的物件移除屬性的所有值。

欄位

屬性名稱

指定屬性的名稱。

類別名稱

(選擇性) 指定目標物件的類別名稱。保留為空白可使用目前物件的類別名稱。

物件

選取目標物件。此物件可以是目前物件，也可以由 DN 或關聯指定。

範例

執行 清除來源屬性值

輸入屬性名稱:* Member

輸入類別名稱:

選取物件: DN

輸入 DN:* "Users\ManagerGroup"

3.6.13 由 XPath 運算式複製

將 XPath 運算式所選取的 XML 節點集之深層副本，附加到另一個 XPath 運算式所選取的元素集。

欄位

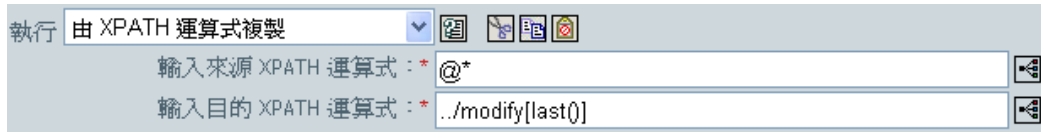
來源 XPATH 運算式

指定傳回節點集 (包含要複製的節點) 的 XPath 1.0 運算式。

目的 XPATH 運算式

指定傳回節點集 (包含要附加所複製節點的元素) 的 XPath 1.0 運算式。

範例



3.6.14 複製操作屬性

將目前操作內出現的所有屬性複製到目前操作內的不同屬性。

欄位

來源名稱

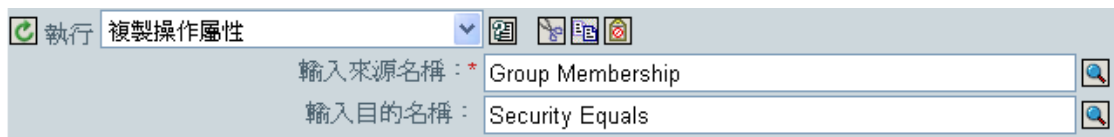
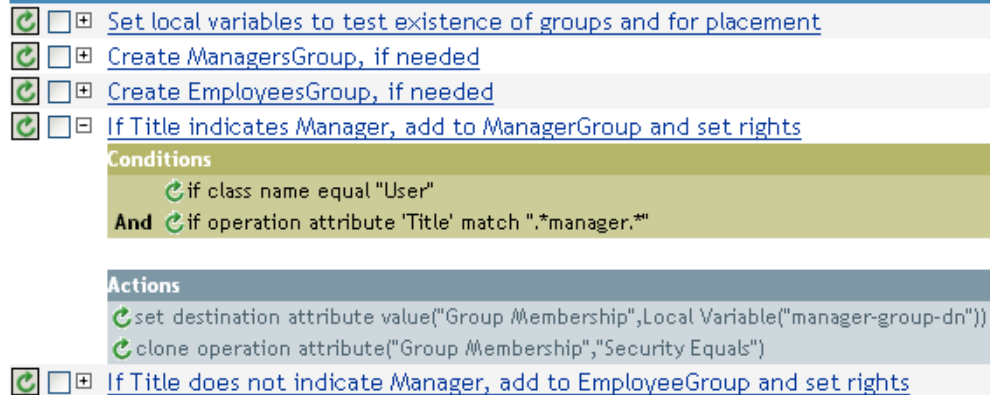
指定複製來源的屬性名稱。

目的名稱

指定複製目的的屬性名稱。

範例

此範例會依據 Title，將「使用者」物件新增至適當的群組 (Employee 或 Manager)。同時也會視需要建立群組，並設定與該群組相等的安全性。該項規則為「依 Title 屬性管理使用者的群組」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。



Clone Operation 屬性會取得來自 Group Membership 屬性的資訊，並將該資訊新增至 Security Equals 屬性，使其值相同。

3.6.15 刪除目的物件

刪除目的資料儲存中的物件。

欄位

模式

選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。

物件

在目的資料儲存中選取要刪除的目標物件。此物件可以是目前物件，也可以由 DN 或關聯指定。

範例



The screenshot shows a configuration dialog for the 'Delete Destination Object' action. The '執行' (Execute) dropdown is set to '刪除目的物件'. Below it, there are three fields: '選取模式' (Select Mode) is set to '新增至目前操作', '選取物件' (Select Object) is set to 'DN', and '輸入 DN' (Enter DN) is set to '"Users/Fred Flintstone"'. There are also several icons for additional actions and a help icon.

3.6.16 刪除來源物件

刪除來源資料儲存中的物件。

欄位

模式

選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。

物件

在來源資料儲存中選取要刪除的目標物件。此物件可以是目前物件，也可以由 DN 或關聯指定。

範例



The screenshot shows a configuration dialog for the 'Delete Source Object' action. The '執行' (Execute) dropdown is set to '刪除來源物件'. Below it, there are two fields: '選取物件' (Select Object) is set to 'DN', and '輸入 DN' (Enter DN) is set to '"Users/Fred Flintstone"'. There are also several icons for additional actions and a help icon.

3.6.17 尋找相符物件

在目的資料儲存中尋找與目前物件相符的物件。

欄位

範圍

在目的資料儲存中尋找與目前物件相符的物件。

DN

指定搜尋的基礎 DN。

相符屬性

指定要搜尋的屬性值。

備註

「尋找相符物件」僅在目前操作為新增操作時有效。

當範圍為「項目」時需要使用 DN 引數，其他範圍則可選擇性使用。當範圍為「子網路樹」或「從屬物件」時，至少需要有一個相符屬性。如果範圍為項目，並且已指定相符屬性，則結果會不明確。如果目的資料儲存為已連接應用程式，則會在目前操作中為所傳回的每個成功相符項目新增關聯。若目前操作已擁有非空白關聯，則不會執行查詢，因此可允許多個尋找相符物件動作串接在相同規則中。

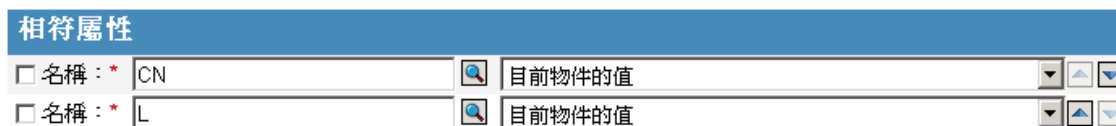
若目的資料儲存為 Identity Vault，則會設定目前操作的目的 DN 屬性。若目前操作已擁有非空白的目的 DN 屬性，則不會執行查詢，因此可允許多個尋找相符物件動作串接在相同規則中。若僅傳回單一結果，並且其尚未關聯，則目前操作的目的 DN 會設定為相符物件的來源 DN。若僅傳回單一結果，並且其已關聯，則目前操作的目的 DN 會設定為單一字元 �。若傳回多個結果，則目前操作的目的 DN 會設定為單一字元 。

範例

此範例會比對有屬性 CN 和 L 的「使用者」物件。規則搜尋的位置會從 Users 容器開始，並且將儲存在 OU 屬性內的資訊新增至 DN。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「比對：依屬性值」，第 87 頁。



當您按一下「引數產生器」圖示時，便會啟動「相符屬性產生器」。您可以在產生器中指定想要比對的屬性。此範例是使用 CN 和 L 屬性。



3.6.18 針對每個

對節點集內的每個節點重複一組動作。

欄位

節點集

指定節點集。

動作

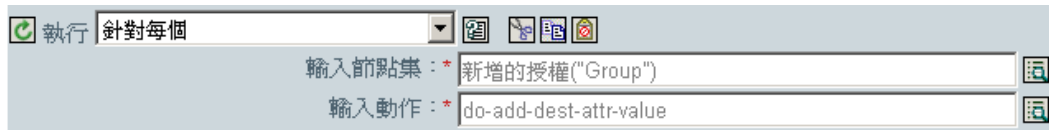
指定要在節點集內的每個節點上執行的動作。

備註

若使用本地變數，則每重複一次動作，目前節點的值都不同。

如果節點集內的某個節點為授權，則「針對每個」會以隱含方式執行「實作授權」，第 262 頁動作。

範例



以下為用來提供動作引數的「引數動作產生器」範例：



3.6.19 產生事件

傳送使用者定義的事件至 Novell Audit。

欄位

ID

事件的 ID。提供的值必須在使用 `java.lang.Integer` 的 `parseInt` 方法剖析時，產生 1000-1999 之間的整數。

層級

事件的層級。

層級	描述
log-emergency	導致 Metadirectory 引擎或驅動程式關閉的事件。
log-alert	需要立即注意的事件。

層級	描述
log-critical	可導致部份 Metadirectory 引擎或驅動程式失效的事件。
log-error	描述可由 Metadirectory 引擎或驅動程式處理之錯誤的事件。
log-warning	不代表問題的負面事件。
log-notice	管理員可用來瞭解或改善使用和操作的事件 (正面或負面事件)。
log-info	任何重要的正面事件。
log-debug	可讓支援人員或工程師進行 Metadirectory 引擎或驅動程式操作除錯的相關事件。

字串

指定要包含在事件內之使用者定義的字串、整數和二進位值。這些值是使用「具名字串產生器」來提供。

標籤	描述
target	將對其採取動作的物件。
target-type	為目標指定預先定義格式的整數。 target-type 的預先定義值目前為： <ul style="list-style-type: none"> ◆ 0 = 無 ◆ 1 = 斜線標記 ◆ 2 = 點標記 ◆ 3 = LDAP 標記
subTarget	將對其採取動作之目標的子元件。
text1	在此處輸入的文字會儲存在 text1 事件欄位中。
text2	在此處輸入的文字會儲存在 text2 事件欄位中。
text3	在此處輸入的文字會儲存在 text3 欄位中。
value	在此處輸入的任何數字都會儲存在 value 事件欄位中。
value3	在此處輸入的任何數字都會儲存在 value3 事件欄位中。
data	在此處輸入的資料會儲存在 blob 事件欄位中。

備註

Novell Audit 事件結構包含一個 **target**、一個 **subTarget**、三個字串 (**text1**、**text2**、**text3**)、兩個整數 (**value**、**value3**)，以及一個一般欄位 (**data**)。文字欄位限制為 256 個位元組，同時 **data** 欄位最多可以包含 3 KB 的資訊，除非在您的環境中啓用了較大的 **data** 欄位。

範例

此範例有四個規則 (Rule)，這些規則根據 **Surname** 屬性的第一個字元實作「使用者」物件的「佈置」規則 (Policy)，並產生追蹤訊息和自定的 Novell Audit 事件。「產生事件」動作

用於傳送事件至 Novell Audit。規則名稱爲「依姓氏放置的規則」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「可下載的 Identity Manager 規則」，第 32 頁。

The screenshot shows a rule configuration window with the following details:

- Setup Local Variables** (expanded)
- Surname A-I: place in Users1** (expanded)
 - Conditions**
 - if class name equal "User"
 - And if operation attribute 'Surname' match "[a-i].*"
 - Actions**
 - set operation destination DN(dn("Training\Users\Active\Users1"+"\"+Operation Attribute("CN")))
 - trace message(color="yellow",Local Variable("LVUsers1"))
 - generate event(id="1000",text1=Local Variable("LVUsers1"))
- Surname J-R: place in Users2** (expanded)
- Surname S-Z: place in Users3** (expanded)

The '產生事件' (Generate Event) dialog box contains the following fields:

- 輸入 ID : * 1000
- 選取層級 : 資訊
- 輸入字串 : text1

以下爲用來提供字串引數的「具名字串產生器」範例：

The '字串' (String) configuration dialog box shows:

- 名稱 : * text1
- 字串值 : * 本地變數("LVUsers1")

「產生事件」會建立具有 ID 1000 的事件，並顯示由 LVUser1 的本地變數所產生的文字。本地變數 LVUser1 是新增至 "+"Training\Users\Active\Users1"+" container" 的 User:Operation Attribute "cn" +" 字串。此事件會讀取新增至 Training\Users\Active\Users1 容器的 User:jsmith。

3.6.20 實作授權

指定實作授權的動作，使那些授權的狀態可以回報給獲授予或遭撤銷授權的代辦。

欄位

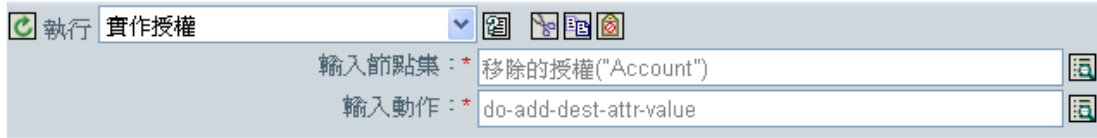
節點集

包含由指定動作實作之授權的節點集。

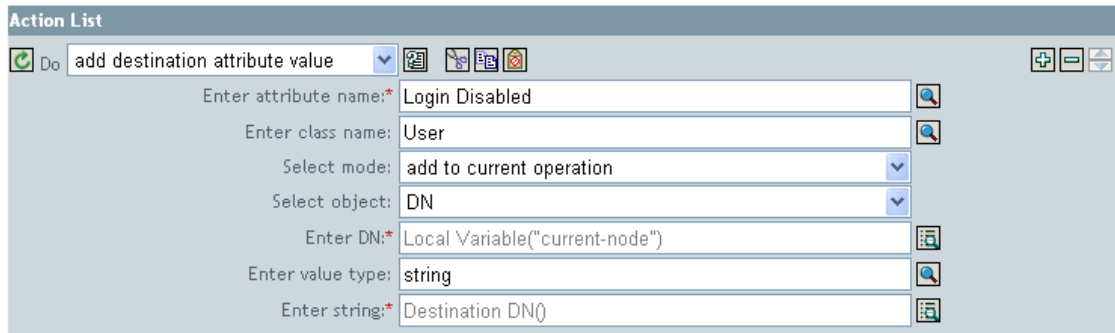
動作

實作指定授權的動作。

範例



以下為用來提供動作引數的「引數動作產生器」範例：



3.6.21 移動目的物件

移動目的資料儲存中的物件。

欄位

模式

選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。

類別名稱

(選擇性) 指定要移動之物件的類別名稱。保留為空白可使用目前物件的類別名稱。

要移動的物件

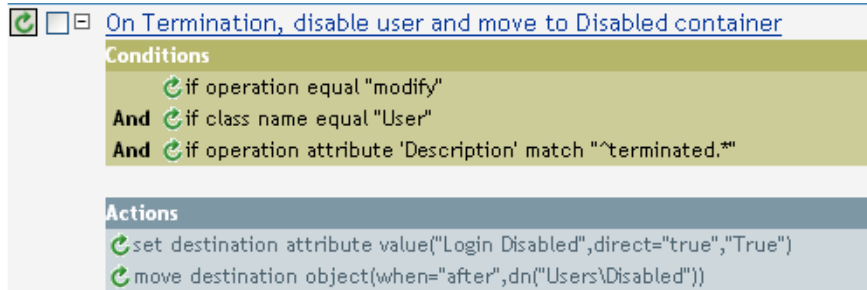
選取要移動的物件。此物件可以是目前物件，也可以由 DN 或關聯指定。

容器

選取要接收物件的容器。此容器可以由 DN 或關聯指定。

範例

此範例包含單一規則，此規則會在 **Description** 屬性指出使用者的帳戶為已終止時，停用該帳戶並移至已停用容器。此規則名為「在終止時停用並移動使用者帳戶」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「可下載的 **Identity Manager** 規則」，第 32 頁。



此規則會檢查它是否為「使用者」物件上的修改事件，以及 **Description** 屬性是否包含已終止的值。若情況如此，它會將 **Login Disabled** 的屬性設定為 **true**，並將物件移入 **User\Disabled** 容器中。

3.6.22 移動來源物件

移動來源資料儲存中的物件。

欄位

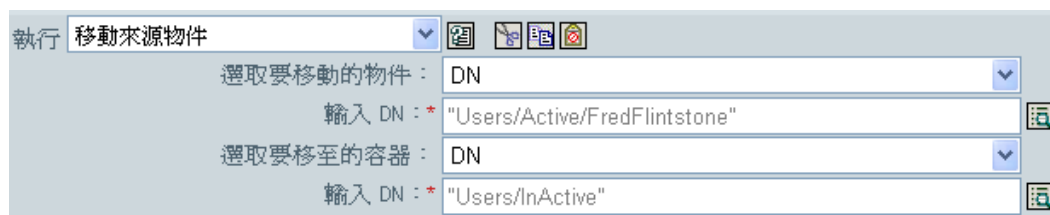
要移動的物件

選取要移動的物件。此物件可以是目前物件，也可以由 **DN** 或關聯指定。

選取容器

選取要接收物件的容器。此容器可以由 **DN** 或關聯指定。

範例



3.6.23 重新格式化操作屬性

使用某模式，重新格式化目前操作內的所有屬性值。

欄位

名稱

指定屬性的名稱。

值類型

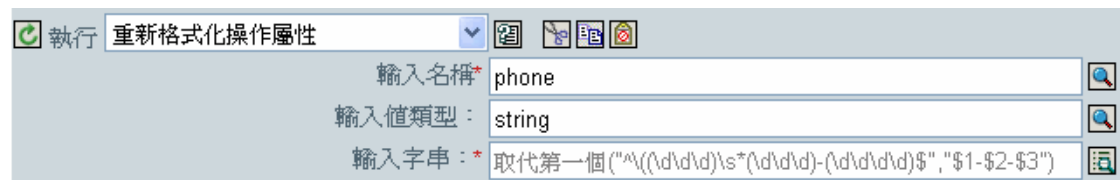
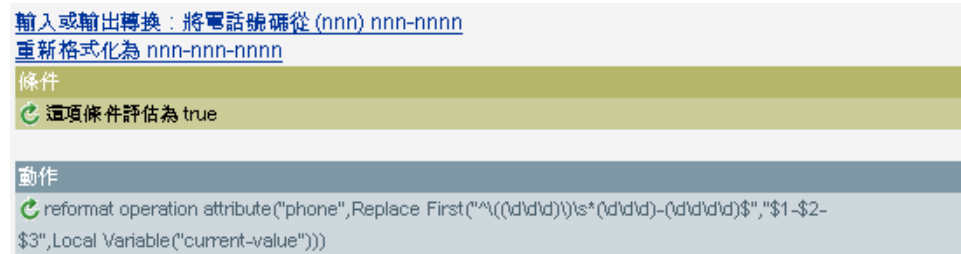
指定新屬性值的語法。

值

指定要用來做為屬性值之新格式的模式值。若需要原始值來建構新值，則必須藉由參考本地變數 `current-value` 來取得該值。

範例

此範例會重新格式化電話號碼。電話號碼會由 `(nnn)-nnn-nnnn` 變更為 `nnn-nnn-nnnn`。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[輸入或輸出轉換：將「電話號碼」從 \(nnn\) nnn-nnnn 重新格式化為 nnn-nnn-nnnn](#)」，第 218 頁。



重新格式化操作屬性動作會變更電話號碼的格式。此規則會使用「引數產生器」及一般運算式來變更顯示資訊的方式。

3.6.24 移除關聯

傳送移除關聯指令至 Identity Vault。

欄位

模式

選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。

關聯

指定要移除之關聯的值。

範例

此範例會採取刪除操作並以停用「使用者」物件做為取代。並進行事件的轉換。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[指令轉換：要停用的發行者刪除](#)」，第 212 頁。

指令轉換：要停用的發行者刪除

The screenshot displays a rule configuration window with the following sections:

- 條件 (Conditions):**
 - 如果操作等於 delete
 - 或 如果類別名稱等於 User
- 動作 (Actions):**
 - set destination attribute value("Login Disabled","true")
 - remove association(association(Association()))
- 執行 (Execution):**
 - 執行 移除關聯
 - 選取模式：新增至目前操作
 - 輸入關聯：* 關聯()

在對「使用者」物件進行刪除操作時，Login Disabled 屬性的值會被設定為 true，並且會移除與物件的關聯。由於已連接應用程式內的關聯物件已不存在，因此會移除關聯。

3.6.25 移除目的屬性值

從目的資料儲存中的物件移除屬性值。

欄位

屬性名稱

指定屬性的名稱。

類別名稱

(選擇性) 指定目標物件的類別名稱。保留為空白可使用目前物件的類別名稱。

模式

選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。

選取物件。

選取目標物件。此物件可以是目前物件，也可以由 DN 或關聯指定。

值類型

指定新屬性值的語法。

值

指定新屬性的值。

範例

執行	移除目的屬性值	☰	🔍	🗑️	📄	🔍
輸入屬性名稱 *	Member					🔍
輸入類別名稱						🔍
選取模式	新增至目前操作					▼
選取物件	DN					▼
輸入 DN *	"Users/ManagerGroup"					🔍
輸入值類型	string					🔍
輸入字串 *	目的 DN()					🔍

3.6.26 移除來源屬性值

從來源資料儲存中物件上的具名屬性移除指定的值。

欄位

屬性名稱

指定屬性的名稱。

類別名稱

(選擇性) 指定目標物件的類別名稱。保留為空白可使用目前物件的類別名稱。

物件

選取目標物件。此物件可以是目前物件，也可以由 DN 或關聯指定。

值類型

指定要移除之屬性值的語法。

值

指定要移除的屬性值。

範例

執行	移除來源屬性值	☰	🔍	🗑️	📄	🔍
輸入屬性名稱 *	Member					🔍
輸入類別名稱						🔍
選取物件	DN					▼
輸入 DN *	"Users/ManagerGroup"					🔍
輸入值類型	string					🔍
輸入字串 *	來源 DN()					🔍

3.6.27 重新命名目的物件

將目的資料儲存中的物件重新命名。

欄位

模式

選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。

物件

選取目標物件。此物件可以是目前物件，也可以由 DN 或關聯指定。

字串

指定物件的新名稱。

範例



The screenshot shows a configuration window titled "執行 重新命名目的物件". It contains the following fields:

- 選取模式: 新增至目前操作
- 選取物件: DN
- 輸入 DN: "Users/Active/Fred Flintstone"
- 輸入字串: "Freddy"

3.6.28 重新命名操作屬性

將目前操作中出現的所有屬性重新命名。

欄位

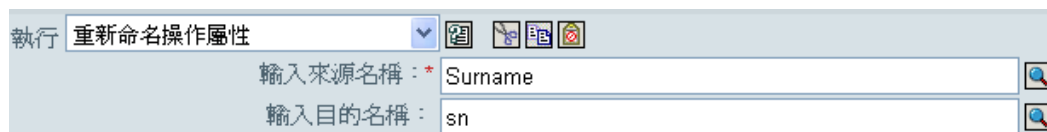
來源名稱

指定原始的屬性名稱。

目的名稱

指定新的屬性名稱。

範例



The screenshot shows a configuration window titled "執行 重新命名操作屬性". It contains the following fields:

- 輸入來源名稱: Surname
- 輸入目的名稱: sn

3.6.29 重新命名來源物件

將來源資料儲存中的物件重新命名。

欄位

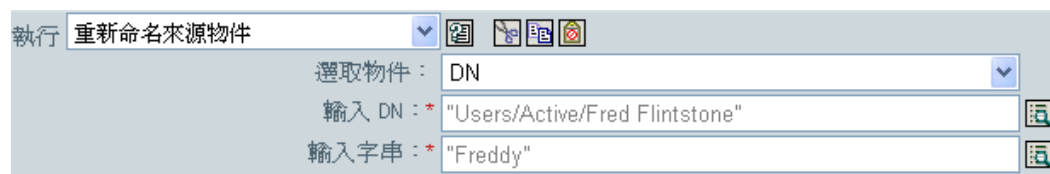
選取物件

選取目標物件。此物件可以是目前物件，也可以由 DN 或關聯指定。

字串

指定物件的新名稱。

範例



3.6.30 傳送電子郵件

傳送電子郵件通知。

欄位

ID

(選擇性) 指定 SMTP 系統中傳送訊息的「使用者 ID」。

伺服器

指定 SMTP 伺服器名稱。

密碼

(選擇性) 指定 SMTP 伺服器帳戶密碼。

重要：密碼屬性的值會以純文字儲存。

類型

選取電子郵件訊息類型。

字串

指定包含各種電子郵件地址、主旨和訊息的值。下表列出有效的具名字串引數：

字串名稱	描述
to	新增地址到電子郵件收件者的清單中；允許使用多個例項。
cc	新增地址到 CC (副本) 電子郵件收件者的清單中；允許使用多個例項。
bcc	新增地址到 BCC (密件副本) 電子郵件收件者的清單中；允許使用多個例項。
from	指定要做為寄出電子郵件地址的地址。
reply-to	指定要做為電子郵件訊息回覆地址的地址。
subject	指定電子郵件主旨。
message	指定電子郵件訊息的內容。
encoding	指定電子郵件訊息使用的字元編碼。

範例



以下為用來提供字串引數的「具名字串產生器」範例：

字串			
<input type="checkbox"/> 名稱：*	to	字串值：*	"to_user1@company.com"
<input type="checkbox"/> 名稱：*	to	字串值：*	"to_user@company.com"
<input type="checkbox"/> 名稱：*	cc	字串值：*	"cc_user@company.com"
<input type="checkbox"/> 名稱：*	bcc	字串值：*	"bcc_user@company.com"
<input type="checkbox"/> 名稱：*	from	字串值：*	"from_user@company.com"
<input type="checkbox"/> 名稱：*	subject	字串值：*	"This is the e-mail subject"
<input type="checkbox"/> 名稱：*	message	字串值：*	"This is the e-mail body"

3.6.31 從範本傳送電子郵件

使用範本產生電子郵件通知。

欄位

通知 DN

以斜線形式指定 SMTP 通知組態物件的 DN。

範本 DN

以斜線形式指定電子郵件範本物件的 DN。

密碼

(選擇性) 指定 SMTP 伺服器帳戶密碼。

重要：密碼屬性的值會以純文字儲存。

字串

指定電子郵件訊息的其他欄位。下列表格包含指定各種電子郵件地址的保留欄位名稱：

字串名稱	描述
to	新增地址到電子郵件收件者的清單中；允許使用多個例項。
cc	新增地址到 CC (副本) 電子郵件收件者的清單中；允許使用多個例項。
bcc	新增地址到 BCC (密件副本) 電子郵件收件者的清單中；允許使用多個例項。
reply-to	指定要做為電子郵件訊息回覆地址的地址。

字串名稱	描述
encoding	指定電子郵件訊息使用的字元編碼。

各個範本也會定義電子郵件訊息的主旨和本文中，可以被取代的欄位。

範例

執行 從範本傳送電子郵件

輸入通知 DN : * /cn=security/cn=Default Notification Collection

輸入範本 DN : * /cn=security/cn=Default Notification Collection/cn=PS-Syr

輸入密碼 :

輸入字串 : manager,surname,given-name,to,cc

以下為用來提供字串引數的「具名字串產生器」範例：

字串	
<input type="checkbox"/> 名稱 : *	manager 字串值 : "Bill Jones"
<input type="checkbox"/> 名稱 : *	surname 字串值 : "Smith"
<input type="checkbox"/> 名稱 : *	given-name 字串值 : "Joe"
<input type="checkbox"/> 名稱 : *	to 字串值 : "to_user@company.com"
<input type="checkbox"/> 名稱 : *	cc 字串值 : "cc_user@company.com"

3.6.32 設定預設的屬性值

如果屬性尚未有值，則對目前操作（以及選擇性地對來源資料儲存中的目前物件）新增預設值。此動作僅在目前操作為新增操作時有效。

欄位

屬性名稱

指定預設屬性的名稱。

寫回

選擇是否也要將預設值寫回來源資料儲存。

值

指定屬性的預設值。

範例

此範例會為「公司」屬性設定預設值。您可以為您選擇的屬性設定值。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[建立：設定預設屬性值](#)」，第 214 頁。

The screenshot displays the configuration interface for a rule named "設定預設屬性值" (Set default attribute value). The rule is categorized under "條件" (Conditions) with the condition "如果類別名稱等於 User" (If class name equals User). The action is "Set default attribute value ([輸入屬性名稱],[write-back=true],[輸入預設屬性值])".

The configuration details are as follows:

- 執行** (Execute): 設定預設屬性值
- 輸入屬性名稱** (Input attribute name): company
- 寫回** (Write back): true
- 輸入引數值** (Input argument value): "Digital Airlines"

The "引數值" (Argument value) section is also visible, showing:

- 類型** (Type): string
- 輸入字串** (Input string): "Digital Airline Inc"

為了建構值，「引數值清單產生器」會被啟動。如需產生器的相關資訊，請參閱「[引數值清單產生器](#)」，第 205 頁。您可以將值設定為所需的值。在此例中，我們使用「引數產生器」，並且將文字設定為公司的名稱。

3.6.33 設定目的屬性值

為目的資料儲存中物件上的屬性新增值，並且移除該屬性的所有其他值。

欄位

屬性名稱

指定屬性的名稱。

類別名稱

(選擇性) 指定目的資料儲存中目標物件的類別名稱。保留為空白可使用目前物件的類別名稱。

模式

選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。

物件

選取目標物件 此物件可以是目前物件，也可以由 DN 或關聯指定。

值類型

選取要設定之屬性值的語法。

值

指定要設定的屬性值。

範例

此範例會採取刪除操作並以停用「使用者」物件做為取代。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[指令轉換：要停用的發行者刪除](#)」，第 212 頁。









指令轉換：要停用的發行者刪除

條件

- 如果操作等於 delete
- 或 如果類別名稱等於 User

動作

- set destination attribute value("Login Disabled","true")
- remove association(association(Association()))

執行	設定目的屬性值	   
輸入屬性名稱：	<input type="text" value="Login Disabled"/>	
輸入類別名稱：	<input type="text"/>	
選取模式：	<input type="text" value="新增至目前操作"/>	
選取物件：	<input type="text" value="目前物件"/>	
輸入值類型：	<input type="text" value="string"/>	
輸入字串：	<input type="text" value="true"/>	

此規則會將 Login Disabled 的屬性值設定為 true。此規則會使用「引數產生器」為屬性值新增 true 文字。如需產生器的相關資訊，請參閱「[引數產生器](#)」，第 203 頁。

3.6.34 設定目的密碼

為目的資料儲存中的目前物件設定密碼。

欄位

模式

選擇此動作應被新增至目前操作之前或之後，或直接寫入目的資料儲存。

物件

選取目標物件。此物件可以是目前物件，也可以由 DN 或關聯指定。

字串

指定要設定的密碼。

範例

此範例會為建立的「使用者」物件設定預設的密碼。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[建立：設定預設密碼](#)」，第 215 頁。

The screenshot shows a rule configuration interface. At the top, the rule name is "建立：設定預設密碼". Under the "條件" (Conditions) section, there is a single condition: "如果類別名稱等於 User". Under the "動作" (Actions) section, there is a single action: "set destination password(Attributer("Given Name")+Attribute("Surname"))". Below this, there is a "執行" (Execute) section with a dropdown menu set to "設定目的密碼". To the right of the dropdown are icons for help, undo, redo, and delete. Below the dropdown, there are two fields: "選擇模式" (Select Mode) set to "新增至目前操作" and "輸入字串" (Input String) set to "屬性("Given Name")+屬性("Surname")".

當建立「使用者」物件時，其密碼會被設定為 Given Name 屬性加上 Surname 屬性的組合。

3.6.35 設定本地變數

設定本地變數。

欄位

變數名稱

指定新本地變數的名稱。

變數類型

選取本地變數的類型。此類型可以為字串、XPath 1.0 節點集或 Java 物件。

範例

此範例會依據 Title，將「使用者」物件新增至適當的群組 (Employee 或 Manager)。同時也會視需要建立群組，並設定與該群組相等的安全性。此規則名稱為「依 Title 管理使用者的群組」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

The screenshot shows a rule configuration interface for the rule "Set local variables to test existence of groups and for placement". Under the "條件" (Conditions) section, there are three conditions: "如果類別名稱等於 'User'", "如果操作等於 'add'", and "如果操作等於 'modify'". The first condition is connected to the second by "且" (AND), and the second is connected to the third by "或" (OR). Under the "動作" (Actions) section, there are four actions: "設定本地變數('manager-group-dn','Users\ManagersGroup')", "設定本地變數('manager-group-info',目的屬性('Object Class',dn(本地變數('manager-group-dn'))))", "設定本地變數('employee-group-dn','Users\EmployeesGroup')", and "設定本地變數('employee-group-info',目的屬性('Object Class',dn(本地變數('employee-group-dn'))))".

本地變數會設定為「使用者」物件的目的屬性（屬於「物件類別」）中的值加上 manager-group-info 的「本地變數」。使用了「引數產生器」建構本地變數。如需相關資訊，請參閱「引數產生器」，第 203 頁。

3.6.36 設定操作關聯

為目前操作設定關聯值。

欄位

關聯

提供新的關聯值。

範例

3.6.37 設定操作類別名稱

為目前操作設定物件類別名稱。

欄位

字串

指定新的類別名稱。

範例

3.6.38 設定操作目的 DN

為目前操作設定目的 DN。

欄位

DN

指定新的目的 DN。

範例

此範例會使用從已連接系統鏡像複製的結構，將物件放置在 Identity Vault 中。您必須定義鏡像複製在來源和目的資料儲存中的哪一點開始。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[建立：設定預設屬性值](#)」，第 75 頁。

佈置：鏡像複製發行者

條件

- if source DN in subtree "[輸入來源階層的基礎]"

動作

- set local variable("dest-base","[輸入目的階層的基礎])
- set operation destination DN(dn(Local Variable("dest-base")+"+Unmatched Source DN (convert="true")))

執行 **設定操作目的 DN**

輸入 DN : * 本地變數("dest-base")+不符的來源 DN(轉換="true")

此規則會將操作目的 DN 設定為目的基礎位置的本地變數加上來源 DN。

3.6.39 設定操作內容

設定操作內容。操作內容是儲存在操作內的具名值。它通常用來提供處理操作結果之規則可能會需要的其他上下文。

欄位

內容名稱

指定操作內容的名稱。

字串

指定操作內容的名稱。

範例

執行 **設定操作內容**

輸入內容名稱 : * myStoredProperty

輸入字串 : * token-string()

3.6.40 設定操作來源 DN

為目前操作設定來源 DN。

欄位

DN

指定新的來源 DN。

範例



3.6.41 設定操作範本 DN

將目前操作的範本 DN 設定為指定的值。此動作僅在目前操作為新增操作時有效。

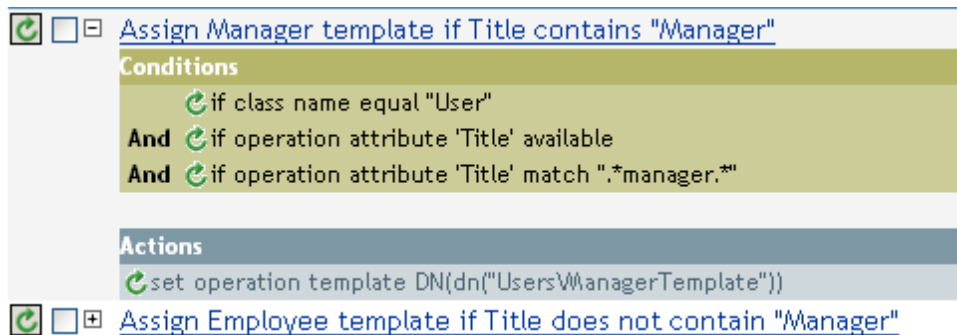
欄位

DN

指定範本 DN。

範例

若 Title 屬性包含 Manager 一字，此範例會套用 Manager 範本。此規則的名稱為「規則：依 Title 指定使用者的範本」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。



範本 Manager Template 會被套用至任何有 Title 屬性可用，且 Title 中含有 manager 一字的「使用者」物件。此規則會使用一般運算式尋找所有可能相符的項目。

3.6.42 設定來源屬性值

為來源資料儲存中物件上的屬性新增值，並且移除該屬性的所有其他值。

欄位

屬性名稱

指定屬性的名稱。

類別名稱

(選擇性) 指定來源資料儲存中目標物件的類別名稱。保留為空白可使用目前物件的類別名稱。

物件

選取目標物件。此物件可以是目前物件，也可以由 DN 或關聯指定。

值類型

選取屬性值的語法。

值

指定要設定的屬性值。

範例

此範例會偵測電子郵件地址的變更，並且將其設回其原始值。規則名稱為「規則：重設電子郵件屬性值」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

[Push back in email changing](#)

條件

- 如果 類別名稱 等於 "User"
- 且** 如果 操作屬性 'Email' 變更

動作

- 設定來源屬性值("Email",目的屬性("Internet EMail Address"))
- 條紋化操作屬性("Email")

執行 設定來源屬性值

輸入屬性值：* Email

輸入類別名稱：

選取物件：目前物件

輸入值類型：string

輸入字串：* 目的屬性("Internet EMail Address")

此動作會取目的屬性 Internet EMail Address 的值，並將「電子郵件」的來源屬性設定為這個值。

3.6.43 設定來源密碼

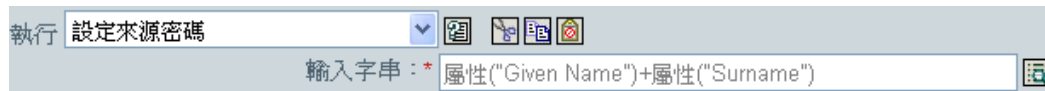
為來源資料儲存中的目前物件設定密碼。

欄位

字串

指定要設定的密碼。

範例



3.6.44 設定 SSO 身分證明

在建立使用者物件或修改密碼時，設定 SSO 身分證明。此動作是「身分證明提供」規則的一部份。如需相關資訊，請參閱第 4 章「Novell 身分證明提供規則」，第 307 頁。

欄位

身分證明儲存物件 **DN**

指定儲存機制物件的 DN。

目標使用者 **DN**

指定目標使用者的 DN。

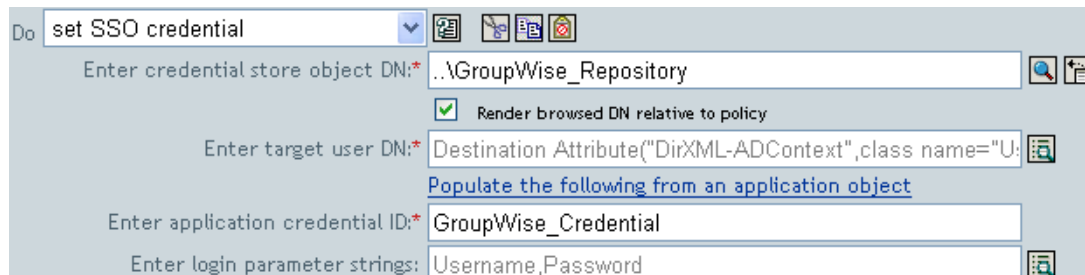
應用程式身分證明 **ID**

指定儲存在應用程式物件中的應用程式身分證明。

登入參數字串

指定應用程式的每一個登入參數。登入參數是儲存在應用程式物件中的驗證金鑰。

範例



3.6.45 設定 SSO 通關密語

設定在提供「使用者」物件時的 Novell SecureLogin® 通關密語和回答。此動作是「身分證明提供」規則的一部份。如需相關資訊，請參閱第 4 章「Novell 身分證明提供規則」，第 307 頁。

欄位

身分證明儲存物件 **DN**

指定儲存機制物件的 DN。

目標使用者 **DN**

指定目標使用者的 DN。

問題與回答字串

指定 SecureLogin 通關密語的問題和回答。

範例


Do set SSO passphrase

Enter credential store object DN:* ..\GroupWise_Repository

Render browsed DN relative to policy

Enter target user DN:* Destination Attribute("DirXML-ADContext",class name="U:

Enter question and answer strings:* "Employee code?",Attribute("workforceID")

SecureLogin 通關密語的問題和回答會以字串形式儲存在規則中。按一下「編輯這些字串」圖示 ，以啟動字串產生器。指定通關密語的問題和回答。

Credential passphrase information is specified by two string elements. The first string contains the question and the second string contains the answer.

Strings * Required

Question:	"Employee Code"
Answer:	Destination Attribute("workforceID",class name="User")

3.6.46 設定 XML 屬性

在 XPath 運算式選取的元素集上設定 XML。

欄位

名稱

指定 XML 屬性的名稱。若先前已在此規則中定義字首，則此名稱可以包含名稱空間字首。

XPATH 運算式

傳回節點集 (包含其上應設定 XML 屬性的元素) 的 XPath 1.0 運算式。

字串

指定 XML 屬性的值。

範例

執行 設定 XML 屬性

輸入名稱* cert-id

輸入 XPATH 運算式: *

輸入字串: "c:\lotus\domino\data\eng.id"

執行 設定 XML 屬性

輸入名稱* cert-pwd

輸入 XPATH 運算式: *

輸入字串: "certify2eng"

3.6.47 設定 SSO 身分證明

在建立使用者物件或修改密碼時，設定 SSO 身分證明。此動作是「身分證明提供」規則的一部份。如需相關資訊，請參閱第 4 章「Novell 身分證明提供規則」，第 307 頁。

欄位

身分證明儲存物件 **DN**

指定儲存機制物件的 DN。

目標使用者 **DN**

指定目標使用者的 DN。

應用程式身分證明 **ID**

指定儲存在應用程式物件中的應用程式身分證明。

登入參數字串

指定應用程式的每一個登入參數。登入參數是儲存在應用程式物件中的驗證金鑰。

範例

Do set SSO credential

Enter credential store object DN: * Novell\Driver Set\GroupWise\GroupWise_Repository

Render browsed DN relative to policy

Enter target user DN: * Destination Attribute("DirXML-ADContext", class name="User")

[Populate the following from an application object](#)

Enter application credential ID: * GroupWise_Credential

Enter login parameter strings: Username, Password

OK Cancel * Required

3.6.48 狀態

產生狀態通知。

欄位

層級

指定通知的狀態層級。

訊息

使用「引數產生器」提供狀態訊息。

備註

若層級為 `retry`，則規則會立即停止輸入文件的處理程序，並為目前正在處理的事件排定重試程序。

若層級為 `fatal`，則規則會立即停止輸入文件的處理程序，並啓始驅動程式的關閉程序。

若目前操作有一個 `event-id`，則會使用該 `event-id` 進行狀態通知，否則不會回報 `event-id`。

範例



3.6.49 去除操作屬性

從目前操作中去掉出現的所有屬性。

欄位

名稱

指定要去除的屬性名稱。

範例

此範例會偵測電子郵件地址的變更，並且將其設回其原始值。規則名稱為「規則：重設電子郵件屬性值」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

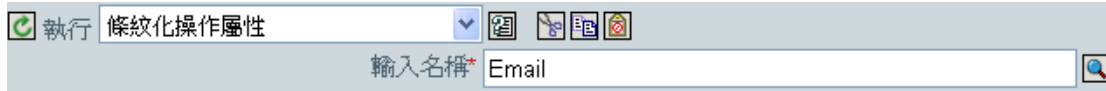
[Push back in email changing](#)

條件

- 如果 類別名稱 等於 "User"
- 且 如果 操作屬性 'Email' 變更

動作

- 設定來源屬性值("Email",目的屬性("Internet EMail Address"))
- 條紋化操作屬性("Email")



此動作會去除 Email 的屬性。會保留的值是原先在目的 Email 屬性中的值。

3.6.50 去除 XPath

去除 XPath 1.0 運算式所選取的節點。

欄位

XPATH 運算式

指定傳回節點集 (包含要去除的節點) 的 XPath 1.0 運算式。

範例



3.6.51 追蹤訊息

傳送訊息到 DSTRACE。

欄位

層級

指定訊息的追蹤層級。預設的層級為 0。只有在指定的追蹤層級小於或等於驅動程式中設定的追蹤層級時，才會出現訊息。

如需如何在驅動程式上設定追蹤層級的相關資訊，請參閱《*Novell Identity Manager 3.0 管理指南*》中的「[檢視 Identity Manager 程序](#)」。

色彩

選取追蹤訊息的色彩。

字串

指定追蹤訊息的值。

範例

此範例有四個規則 (Rule)，這些規則根據 Surname 屬性的第一個字元實作「使用者」物件的「佈置」規則 (Policy)。它會產生追蹤訊息和自定的 Novell Audit 事件。「追蹤訊息」動

作用於傳送追蹤訊息到 DSTRACE 中。規則名稱爲「依姓氏放置的規則」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「可下載的 Identity Manager 規則」，第 32 頁。

Setup Local Variables

Surname A-I: place in Users1

Conditions

- if class name equal "User"
- And if operation attribute 'Surname' match "[a-i]."

Actions

- set operation destination DN(dn("Training\Users\Active\Users1"+"\"+Operation Attribute("CN")))
- trace message(color="yellow",Local Variable("LVUsers1"))
- generate event(id="1000",text1=Local Variable("LVUsers1"))

Surname J-R: place in Users2

Surname S-Z: place in Users3

執行 追蹤訊息

輸入層級：

選取色彩： 黃色

輸入字串： * 本地變數("LVUsers1")

此動作會傳送追蹤訊息到 DSTRACE。本地變數的內容爲 LVUsers1，在 DSTRACE 中會以黃色顯示。

3.6.52 否決

否決目前的操作。

範例

此範例會排除來自指定子網路樹的所有事件。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需預先定義規則的相關資訊，請參閱「事件轉換：範圍過濾：排除子網路樹」，第 217 頁。

事件轉換：範圍過濾：排除子網路樹

條件

- if source DN in subtree"[輸入要排除的子網路樹]"

動作

- veto()

執行 否決

此動作會否決來自指定子網路樹的所有事件。

3.6.53 若無法使用操作屬性則否決

依目前操作中是否有屬性可用，有條件地取消目前的操作及結束目前規則的處理程序。

欄位

名稱

指定屬性的名稱。

範例

此範例不允許建立所有的「使用者」物件，除非可以使用 Given Name、Surname、Title、Description 和 Internet EMail Address 屬性。規則名稱為「強制屬性存在的規則」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

User required attributes: First/Last Name, Title, Description, Email

條件

- 如果 類別名稱 等於 "User"

動作

- 如果操作屬性無法使用則否決("Given Name")
- 如果操作屬性無法使用則否決("Surname")
- 如果操作屬性無法使用則否決("Title")
- 如果操作屬性無法使用則否決("Description")
- 如果操作屬性無法使用則否決("Internet EMail Address")

執行 如果操作屬性無法使用則否決

輸入名稱

如果無法使用 Given Name、Surname、Title、Description 和 Internet Email Address 屬性，此動作會否決該項操作。

3.7 名詞記號

本節包含使用「規則產生器」介面時，可用的所有名詞記號之詳細參考資訊。

- ◆ 「新增的授權」，第 286 頁
- ◆ 「關聯」，第 286 頁
- ◆ 「屬性」，第 287 頁
- ◆ 「類別名稱」，第 287 頁
- ◆ 「目的屬性」，第 287 頁
- ◆ 「目的 DN」，第 288 頁
- ◆ 「目的名稱」，第 289 頁
- ◆ 「授權」，第 290 頁
- ◆ 「全域組態值」，第 290 頁
- ◆ 「本地變數」，第 290 頁
- ◆ 「具名密碼」，第 291 頁
- ◆ 「操作」，第 292 頁
- ◆ 「操作屬性」，第 292 頁

- ◆ 「操作內容」，第 293 頁
- ◆ 「密碼」，第 293 頁
- ◆ 「移除的屬性」，第 293 頁
- ◆ 「Removed Entitlements」，第 293 頁
- ◆ 「Source Attribute」，第 294 頁
- ◆ 「Source DN」，第 294 頁
- ◆ 「Source Name」，第 295 頁
- ◆ 「Text」，第 295 頁
- ◆ 「Unique Name」，第 296 頁
- ◆ 「Unmatched Source DN」，第 297 頁
- ◆ 「XPath」，第 298 頁

3.7.1 新增的授權


展開至目前操作中獲授予的授權值。

欄位

名稱

授權的名稱。

範例

 Added Entitlement("manager")

3.7.2 關聯

展開至目前操作中的關聯值。



範例

此範例取自 Identity Manager 3.0 隨附的預先定義規則。如需預先定義規則的相關資訊，請參閱「[指令轉換：要停用的發行者刪除](#)」，第 212 頁。



「移除關聯」的動作會使用 Association 記號，從目前操作擷取其值。此規則會從「使用者」物件移除關聯，因此任何通過的新事件都不會影響到「使用者」物件。


指令轉換：要停用的發行者刪除

條件

-  如果操作等於 delete
- 或  如果類別名稱等於 User

動作

-  set destination attribute value("Login Disabled","true")
-  remove association(association(Association()))

 Association()

3.7.3 屬性

展開至目前操作和來源資料儲存中，來自目前物件的屬性值。在邏輯上，此屬性可被視為操作屬性記號和來源屬性記號的聯集 (union)。其中不包含從修改操作移除值。

欄位

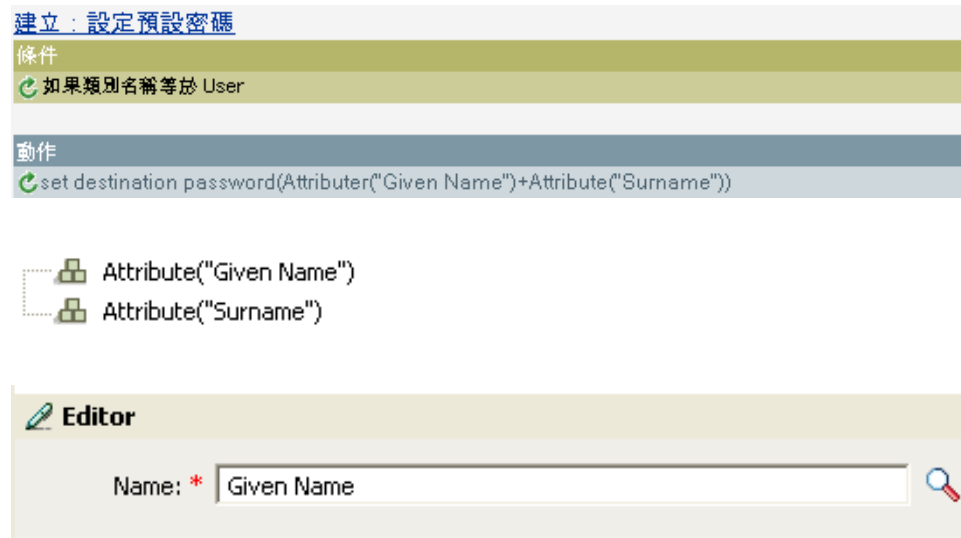
名稱

指定屬性的名稱。

範例

此範例取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[建立：設定預設密碼](#)」，第 215 頁。

「設定目的密碼」的動作使用屬性記號來建立密碼。密碼由 Given Name 屬性和 Surname 屬性所組成。當您在使用「引數產生器編輯器」時，可以瀏覽並選取想要使用的屬性。




The screenshot displays a rule configuration interface. At the top, there is a section titled "建立：設定預設密碼" (Build: Set Default Password). Below this, the "條件" (Condition) section contains the rule "如果類別名稱等於 User" (If class name equals User). The "動作" (Action) section contains the rule "set destination password(Attribute("Given Name")+Attribute("Surname"))". Below the action, two attributes are listed: "Attribute("Given Name")" and "Attribute("Surname")". At the bottom, there is an "Editor" section with a search bar labeled "Name: *" containing the text "Given Name" and a magnifying glass icon.

3.7.4 類別名稱

展開至目前操作中的物件類別名稱。

範例

 Class Name()

3.7.5 目的屬性

展開至目的資料儲存中的目前物件、DN 或關聯之指定屬性值。

欄位

類別名稱

(選擇性) 指定目標物件的類別名稱。保留為空白可使用目前物件的類別名稱。

名稱

屬性名稱。

範例

此範例來自於「依 Title 管理使用者的群組」規則，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

該規則會以「引數產生器」建立 Destination Attribute。「設定本地變數」的動作包含 Destination Attribute 記號。

The screenshot shows a rule configuration window titled "Set local variables to test existence of groups and for placement". It is divided into two main sections: "條件" (Conditions) and "動作" (Actions).
Under "條件", there are three conditions:

- 如果 類別名稱 等於 "User"
- 如果 操作 等於 "add"
- 或 如果 操作 等於 "modify"

The conditions are connected by a "且" (AND) operator.
Under "動作", there are four actions:

- 設定本地變數("manager-group-dn","Users\ManagersGroup")
- 設定本地變數("manager-group-info",目的屬性("Object Class",dn(本地變數("manager-group-dn"))))
- 設定本地變數("employee-group-dn","Users\EmployeesGroup")
- 設定本地變數("employee-group-info",目的屬性("Object Class",dn(本地變數("employee-group-dn"))))

Below the actions, there is a "Destination Attribute" field with the value "Destination Attribute("Object Class", dn())".

The screenshot shows the "Editor" window for configuring a Destination Attribute. It has four input fields:

- 名稱: * 物件類別
- 類別名稱: (empty)
- 選取物件: DN
- 輸入DN: * Local Variable("manager-group-dn")

There are search icons next to the first two fields and a list icon next to the last field.

您可以透過「編輯器」建立 Destination Attribute。在此範例中，設定了「物件類別」的屬性。DN 用於選取物件。DN 的值是 manager-group-dn 的「本地變數」。

3.7.6 目的 DN

展開至目前操作中指定的目的 DN。

欄位

轉換

選擇是否要將 DN 轉換為來源資料儲存所使用的格式。

開始

指定要開始處理的相對可辨識名稱 (Relative Distinguished Name, RDN) 索引：

- ◆ 索引 0 為最根部的 RDN
- ◆ 正索引為最根部之 RDN 的偏移
- ◆ 索引 -1 為最葉部的段落
- ◆ 負索引為從最葉部之 RDN 朝向最根部之 RDN 的偏移

長度

指定要包含的 RDN 數目。負數會解譯為 (段落總數 + 長度) + 1。例如，如果某個 DN 有 5 個段落，長度為 -1，則數目 = (5 + (-1)) + 1 = 5；若長度為 -2，則數目 = (5 + (-2)) + 1 = 4，以此類推。

備註

如果開始和長度設為預設值 {0,-1}，則會使用整個 DN，否則就只會使用開始和長度所指定的部份 DN。

範例

此範例使用 Destination DN 記號設定 target-container 的本地變數值。如果此容器不存在，該規則便會為「使用者」物件建立一個部門容器。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[指令轉換：建立部門容器：第 1 部份和第 2 部份](#)」，第 210 頁。

Command Transformation - Create Departmental Container - Part 1

條件

如果操作等於 add

動作

```
set local variable("target-container",Destination DN(length="-2"))
set local variable("does-target-exist",Destination Attribute("objectclass",class
name="OrganizationalUnit",dn(Local Variable("target-container"))))
```

Destination DN(length="-2")

3.7.7 目的名稱

展開至目前操作中指定之目的 DN 的不合法相對可辨識名稱 (Relative Distinguished Name, RDN)。

範例

Destination Name()

3.7.8 授權

展開至目前物件中獲授予授權的值。

欄位

名稱

授權的名稱。

範例

 Entitlement("manager")

3.7.9 全域組態值


展開至全域組態變數的值。

欄位

名稱

全域組態值的名稱。

範例

 Global Configuration Value("Fred")

3.7.10 本地變數

展開至本地變數的值。

欄位

名稱

指定本地變數的名稱。

範例

此範例來自於「依 Title 管理使用者的群組」規則，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

「新增目的物件」動作會使用 Local Variable 記號。

[Set local variables to test existence of groups and for placement](#)

[Create ManagersGroup, if needed](#)

條件

- 如果 本地變數 'manager-group-info' 可用
- 且 如果 本地變數 'manager-group-info' 不等於 "group"

動作

- 新增目的物件(類別名稱="group",when="before",dn(本地變數("manager-group-dn")))

[Create EmployeesGroup, if needed](#)

[If Title indicates Manager, add to ManagerGroup and set rights](#)

[If Title does not indicate Manager, add to EmployeeGroup and set rights](#)

..... Local Variable("manager-group-dn")

Editor

變數名稱 : *

Local Variables

搜尋 :

- [employee-group-dn](#)
- [employee-group-info](#)
- [fromNds](#)
- [manager-group-dn](#)
- [manager-group-info](#)

「本地變數」只有在先前已於規則中使用「設定本地變數」動作的情形下才能使用。它會設定儲存在「本地變數」中的值。在「編輯器」中，您可以按一下瀏覽圖示和所有列出的已定義本地變數。選取正確的本地變數。

本地變數的值為 `group-manager-dn`。在此規則之前的規則中，「設定位置變數」動作將 `group-manager-dn` 定義為管理員之群組 `Users\ManagersGroup` 的 DN。

3.7.11 具名密碼


展開至驅動程式中的具名密碼。

欄位

名稱

密碼名稱。


範例

 Named Password("password")

3.7.12 操作

展開至目前操作的名稱。

範例

 Operation()

3.7.13 操作屬性

展開至目前操作中的屬性值。其中不包含從修改操作中移除的值。

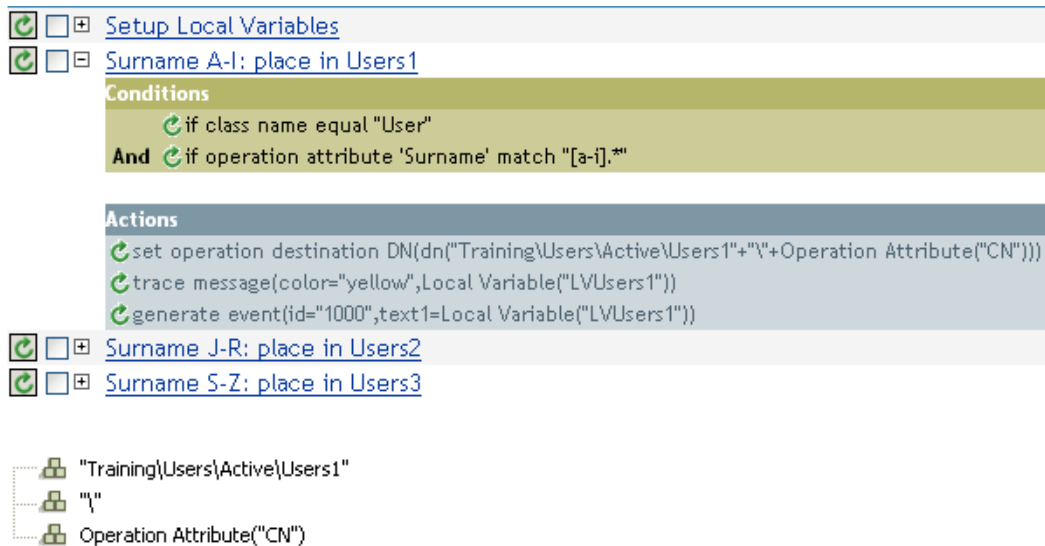
欄位

名稱

指定屬性的名稱。

範例

此範例有四個規則 (Rule)，這些規則根據 Surname 屬性的第一個字元實作「使用者」物件的「佈置」規則 (Policy)。它可以產生追蹤訊息和自定的 Novell Audit 事件。規則名稱為「依姓氏放置的規則」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。






The screenshot displays a configuration window with a list of rules on the left and a detailed view of the selected rule on the right. The rules list includes:

- [Setup Local Variables](#)
- [Surname A-I: place in Users1](#)
- [Surname J-R: place in Users2](#)
- [Surname S-Z: place in Users3](#)

The detailed view for the selected rule shows the following configuration:

- Conditions:**
 - if class name equal "User"
 - And** if operation attribute 'Surname' match "[a-].*"
- Actions:**
 - set operation destination DN(dn("Training\Users\Active\Users1"+"\"+Operation Attribute("CN")))
 - trace message(color="yellow",Local Variable("LVUsers1"))
 - generate event(id="1000",text1=Local Variable("LVUsers1"))

Below the rule details, a tree view shows the object structure:

-  "Training\Users\Active\Users1"
-  "\"
-  Operation Attribute("CN")



「設定操作目的 DN」動作包含 Operation Attribute 記號。Operation Attribute 記號會將「目的 DN」設定為 CN 屬性。此規則會取得 Training\Users\Active\Users 的上下文，並新增 \ 加上 CN 屬性的值。

3.7.14 操作內容

展開至目前操作中指定操作內容的值。

欄位

名稱

指定操作內容的名稱。


範例

 Operation Property("myStoredProperty")

3.7.15 密碼

展開至目前操作中指定的密碼。

範例

 Password()

3.7.16 移除的屬性


展開至目前操作中所指定將被移除的屬性值。此屬性僅會套用至修改操作。

欄位

名稱

指定屬性的名稱。

範例

 Removed Attribute("OU")

3.7.17 Removed Entitlements


展開至目前操作中撤銷的授權值。

欄位

名稱

指定授權的名稱。

範例

 Removed Entitlement("manager")

3.7.18 Source Attribute

展開至來自來源資料儲存內物件的屬性值。

欄位


類別名稱

(選擇性) 指定目標物件的類別名稱。保留為空白可使用目前物件的類別名稱。

名稱

屬性名稱。

範例

 Source Attribute("OU")

3.7.19 Source DN

展開至目前操作中的來源 DN。

欄位

轉換

選取是否要將 DN 轉換為目的資料儲存所使用的格式。

開始

指定要開始處理的 RDN 索引：

- ◆ 索引 0 為最根部的 RDN
- ◆ 正索引為最根部之 RDN 的偏移
- ◆ 索引 -1 為最葉部的段落
- ◆ 負索引為從最葉部之 RDN 朝向最根部之 RDN 的偏移


長度

要包含的 RDN 段落數目。負數會解譯為 (段落總數 + 長度) + 1。例如，如果某個 DN 有 5 個段落，長度為 -1，則數目 = (5 + (-1)) + 1 = 5；若長度為 -2，則數目 = (5 + (-2)) + 1 = 4，以此類推。

備註

如果開始和長度設為預設值 {0,-1}，則會使用整個 DN，否則就只會使用開始和長度所指定的部份 DN。


範例

 Source DN()

3.7.20 Source Name

展開至目前操作中所指定之來源 DN 的不合法相對可辨識名稱 (Relative Distinguished Name, RDN)。

範例

 Source Name()

3.7.21 Text

展開至文字。

欄位

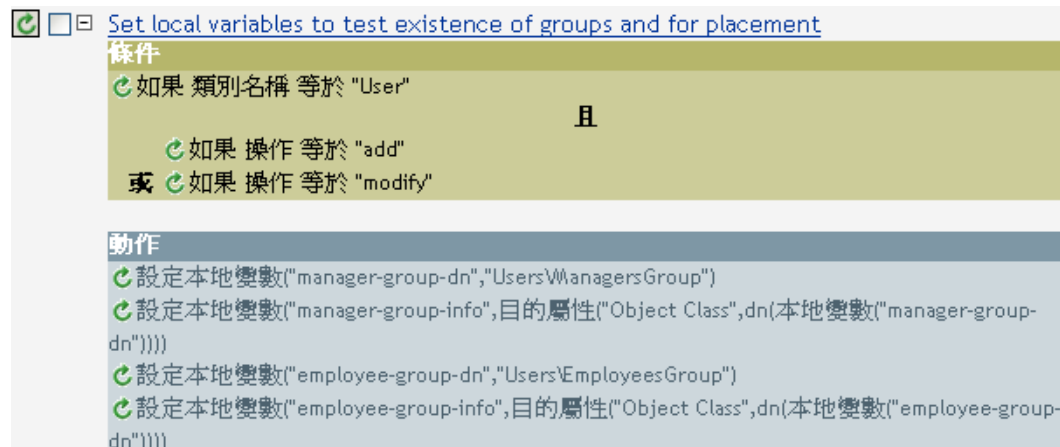
文字

指定文字。

範例

此範例來自於「依 Title 管理使用者的群組」規則，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

Text 記號在「設定位置變數」動作中用來定義管理員的群組 DN。Text 記號可包含物件或純文字。



[Set local variables to test existence of groups and for placement](#)

條件

- 如果 類別名稱 等於 "User"
- 且
- 如果 操作 等於 "add"
- 或 如果 操作 等於 "modify"

動作

- 設定本地變數("manager-group-dn","Users\ManagersGroup")
- 設定本地變數("manager-group-info",目的屬性("Object Class",dn(本地變數("manager-group-dn"))))
- 設定本地變數("employee-group-dn","Users\EmployeesGroup")
- 設定本地變數("employee-group-info",目的屬性("Object Class",dn(本地變數("employee-group-dn"))))

 "Users\ManagersGroup"



Text 記號包含管理員的群組 DN。您可以瀏覽至要使用的物件，或在編輯器中輸入資訊。

3.7.22 Unique Name

展開至模式型名稱；根據指定的準則，該名稱在目的資料儲存中是獨一無二的。

欄位

名稱

指定要檢查其獨特性的屬性名稱。

範圍

指定獨特性的檢查範圍。

開始搜尋

選取搜尋的起始點。起始點可以是資料儲存的根部，也可以由 DN 或關聯指定。

模式

指定要使用的模式，藉由「引數產生器」產生唯一值。

計數器的開始

指定需要尋找唯一名稱時，計數器使用的開始數字。

數字

指定計數器的數字寬度，預設值為 1。開頭核取方塊為 0 的「填補」計數器會預加 0，以符合數字長度。例如，如果數字寬度為 3，起始的唯一值便會先以 001，再以 002 附加上去，以此類推。

備註


就每個提供的模式而言，查詢的執行對象是目的資料儲存，並使用提供的屬性名稱、範圍及搜尋起始進行。每個指定的模式都會依序嘗試使用，直到找到的值未傳回任何找到的物件為止。

如果所有的指定模式都已用盡，便會附加計數器到最後一個模式，且重複嘗試使用該模式（每次增加計數器），直到查詢未傳回任何例項為止。

計數器可設為使用計數器啟動欄位，從不同的數字開始起算。計數器會使用數字欄位所指定的數字數目。如果數字的數目少於指定數目，計數器的右方便會填入零。數字的數目超過指定數目時，則不會產生任何唯一名稱，且所含規則會傳回錯誤狀態。

如果目的資料儲存為 Identity Vault 且名稱欄位保留空白，則會針對虛擬屬性 "[Entry].rdn" 執行搜尋，且該屬性不論命名屬性為何，皆代表物件的相對可辨識名稱 (Relative Distinguished Name, RDN)。如果目的資料儲存為已連接的應用程式，則需要名稱欄位。

範例

 Unique Name("CN",scope="subtree",Lower Case())

以下為建構唯一名稱引數時的「編輯器」窗格範例：



建構了下列模式，可提供唯一名稱：

```
Lower Case()  
├── Substring()  
│   ├── Attribute("Given Name")  
│   └── Attribute("Surname")  
└── +  
    └── Attribute("Surname")
```

如果此模式未產生唯一名稱，則會附加一個數字，並遞增至指定的數字數目。在此範例中，附加的數字會在錯誤發生前，產生九個額外的唯一名稱 (pattern1 - pattern9)。

3.7.23 Unmatched Source DN

展開至目前操作中的部份來源 DN，且其對應於最近的 If Source DN 條件相符項目中，不符合該條件的部份 DN。

欄位

轉換

選取是否轉換目的資料儲存所使用的 DN 格式。

備註

如果沒有相符項目，則會使用整個 DN。

範例

此範例取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[比對：鏡像複製訂閱者：LDAP 格式](#)」，第 220 頁。

「尋找相符物件」動作會使用 Unmatched Source DN 記號，建立輕量目錄存取協定 (Lightweight Directory Access Protocol, LDAP) 格式中的相符資訊。如此便會採用來源 DN 中不相符的部份建立相符項目。

比對：鏡像複製訂閱者：LDAP 格式

條件

```
if source DN in subtree "[輸入來源階層的基礎]"
```

動作

```
set local variable("dest-base","[輸入目的階層的基礎]")
find matching object(scope="entry",dn(Unmatched Source DN(convert="true")+","+Local Variable("dest-base")))
```

Unmatched Source DN(convert="true")
,
Local Variable("dest-base")

Editor

轉換為目的 DN 格式：

3.7.24 XPath

展開至 XPath 1.0 運算式的評估結果。

欄位

運算式

要評估的 XPath 1.0 運算式。

範例

```
XPATCH("*[@attr-name='OU']/value[starts-with(string(.),'xxx']")
```

3.8 動詞記號

本節包含使用「規則產生器」介面時，可用的所有動詞之詳細參考資訊。

- ◆ 「Escape Destination DN」，第 299 頁
- ◆ 「Escape Source DN」，第 299 頁
- ◆ 「Lower Case」，第 299 頁
- ◆ 「Parse DN」，第 300 頁
- ◆ 「Replace All」，第 302 頁
- ◆ 「Replace First」，第 302 頁
- ◆ 「Substring」，第 304 頁

- ◆ 「Upper Case」，第 305 頁

3.8.1 Escape Destination DN

根據目的資料儲存的 DN 格式規則逸出字串。

範例

此範例取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[佈置：一般發行者](#)」，第 91 頁。

「設定目的操作 DN」動作會使用 Escape Destination DN 記號，建立「使用者」物件的目的 DN。

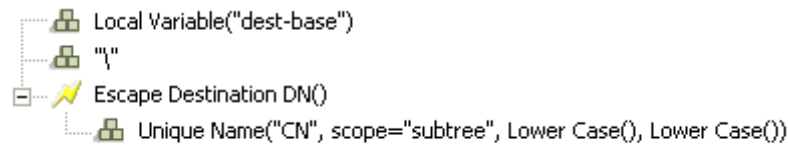
佈置：一般發行者

條件

如果類別名稱等於 User

動作

```
set local variable("dest-base",[輸入目的容器的 DN])
set operation destination DN(dn(Local Variable("dest-base")+""+Escape Destination DN(Unique Name("CN",scope="subtree",Lower Case(Substring(length="1",Operation Attribute("Given Name"))+Operation Attribute("Surname")),Lower Case(Substring(length="2",Operation Attribute("Given Name"))+Operation Attribute("surname"))))))
```



Escape Destination DN 記號會採用「唯一名稱」中的值，並將其設為目的 DN 的格式。

3.8.2 Escape Source DN

根據來源資料儲存的 DN 格式規則逸出字串。

範例

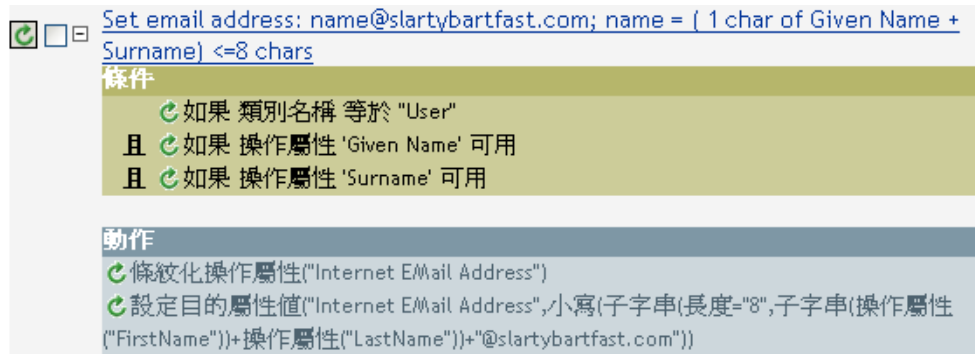
```
Escape Source DN()
  Attribute("Surname")
```

3.8.3 Lower Case

將字串中的字元轉換為小寫。

範例

此範例將電子郵件地址設為 `name@slartybartfast.com`，其中名稱等於「名」的第一個字元加上「姓」。該規則的名稱為「規則：從名和姓建立電子郵件地址」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。



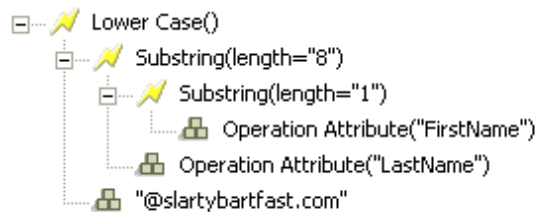
Set email address: name@slartybartfast.com; name = (1 char of Given Name + Surname) <=8 chars

條件

- 如果 類別名稱 等於 "User"
- 且 如果 操作屬性 'Given Name' 可用
- 且 如果 操作屬性 'Surname' 可用

動作

- 條紋化操作屬性("Internet EMail Address")
- 設定目的屬性值("Internet EMail Address",小寫(子字串(長度="8",子字串(操作屬性("FirstName"))+操作屬性("LastName"))+"@slartybartfast.com"))



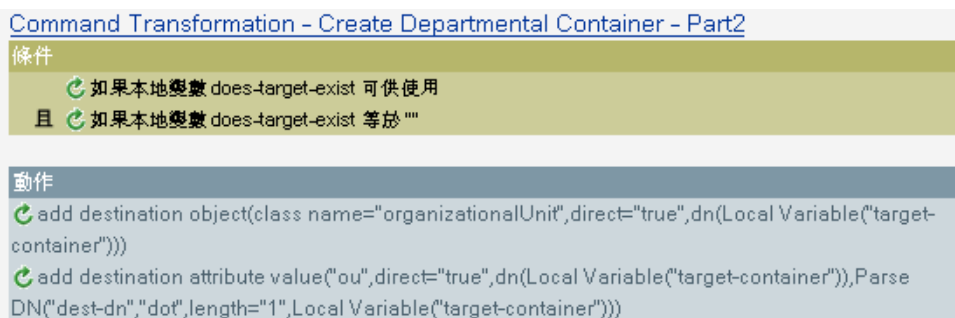
Lower Case 記號會將「設定目的」動作之屬性值中的所有資訊設為小寫。

3.8.4 Parse DN

將 DN 轉換為其他格式。

範例

此範例使用 Parse DN 記號建立「新增目的屬性值」動作的值。此範例取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「[指令轉換：建立部門容器：第 1 部份和第 2 部份](#)」，第 210 頁。



Command Transformation - Create Departmental Container - Part2

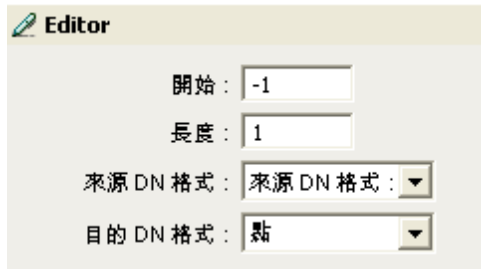
條件

- 如果本地變數 `does-target-exist` 可供使用
- 且 如果本地變數 `does-target-exist` 等於 ""

動作

- add destination object(class name="organizationalUnit",direct="true",dn(Local Variable("target-container")))
- add destination attribute value("ou",direct="true",dn(Local Variable("target-container")),Parse DN("dest-dn","dot",length="1",Local Variable("target-container")))

Parse DN("dest-dn", "dot", length="1", start="-1")
Local Variable("target-container")



Editor

開始： -1

長度： 1

來源 DN 格式： 來源 DN 格式

目的 DN 格式： 點

Parse DN 記號是採用來源 DN 中的資訊，並將其轉換為點標記。Parse DN 中的資訊儲存於 OU 的屬性值中。

欄位

開始

指定要開始處理的相對可辨識名稱 (Relative Distinguished Name, RDN) 索引：

- ◆ 索引 0 為最根部的 RDN
- ◆ 正索引為最根部之 RDN 的偏移
- ◆ 索引 -1 為最葉部的段落
- ◆ 負索引為從最葉部之 RDN 朝向最根部之 RDN 的偏移

長度

要包含的 RDN 數目。負數會解譯為 (段落總數 + 長度) + 1。例如，如果某個 DN 有 5 個段落，長度為 -1，則數目 = (5 + (-1)) + 1 = 5；若長度為 -2，則數目 = (5 + (-2)) + 1 = 4，以此類推。

來源 DN 格式

指定用於解析來源 DN 的格式。

目的 DN 格式

指定用於輸出已解析之 DN 的格式。

來源 DN 分隔符

如果「來源 DN 格式」設為自定，則指定自定來源 DN 分隔符。

目的 DN 分隔符

如果「目的 DN 格式」設為自定，則指定自定目的 DN 分隔符。

備註

如果開始和長度設為預設值 {0,-1}，則會使用整個 DN，否則就只會使用開始和長度所指定的部份 DN。

指定自定 DN 格式時，組成分隔符集的八個字元定義如下：

1、輸入的名稱布林值旗標：0 表示未輸入名稱，1 表示已輸入名稱

2、Unicode 非映射字元布林值旗標：0 表示不要將無法映射的 Unicode 字元輸出或解譯為逸出的十六進位數字字串，例如 \FEFF。eDirectory 不接受下列 Unicode 字元：0xfeff、0xfffe、0xfffd 和 0xffff。

3、相對 RDN 分隔符

4、RDN 分隔符

5、名稱分隔元

6.6、名稱值分隔符

7、萬用字元

8、逸出字元

如果「RDN 分隔符」和「相對 RDN 分隔符」是相同的字元，名稱的走向即為由根部向右，否則即為由根部向左。

如果分隔符號集中的字元超過八個，多餘的字元就會視為需要逸出，但不具有其他特殊意義的字元。

3.8.5 Replace All

取代字串中出現的所有一般運算式。

欄位

一般運算式

指定符合要取代之子字串的一般運算式。

取代為

指定取代字串。

備註

如需建立一般運算式的詳細資料，請參閱：

- ◆ Sun 的 Java 網站 (<http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html>)
- ◆ Sun 的 Java 網站 ([http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#replaceAll\(java.lang.String\)](http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#replaceAll(java.lang.String)))

使用了 CASE_INSENSITIVE、DOTALL 和 UNICODE_CASE 等模式選項，但是可以藉由使用適當的內嵌逸出加以反轉。

範例

```
Replace All("{.}", "$1")  
Destination DN()
```

3.8.6 Replace First

取代字串中首次出現的一般運算式。

欄位

一般運算式

指定符合要取代之子字串的一般運算式。

取代為

指定取代字串。

備註

相符例項會由「取代為」欄位中指定值所指定的字串取代。

如需建立一般運算式的詳細資料，請參閱：

- ◆ Sun 網站 (<http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html>)
- ◆ Sun 的網站 (java.lang.String) ([http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#replaceAll\(java.lang.String\)](http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#replaceAll(java.lang.String)))

使用了 CASE_INSENSITIVE、DOTALL 和 UNICODE_CASE 等模式選項，但是可以藉由使用適當的內嵌逸出加以反轉。

範例

此範例將電話號碼從 (nnn)-nnn-nnnn 重新格式化為 nnn-nnn-nnnn。此規則取自 Identity Manager 3.0 隨附的預先定義規則。如需相關資訊，請參閱「輸入或輸出轉換：將「電話號碼」從 (nnn) nnn-nnnn 重新格式化為 nnn-nnn-nnnn」，第 218 頁。

Replace First 記號用於「重新格式化操作屬性」動作中。

The screenshot displays a configuration window for a rule. At the top, the title is "輸入或輸出轉換：將電話號碼從 (nnn) nnn-nnnn 重新格式化為 nnn-nnn-nnnn". Below the title, there is a "條件" (Condition) section with a green checkmark and the text "這項條件評估為 true". The "動作" (Action) section contains a "reformat operation attribute" with the following configuration: "phone", "Replace First", and a regular expression: `^(\\d\\d\\d)\\s*(\\d\\d\\d)-(\\d\\d\\d\\d)$`. Below this, there is a "Local Variable" section with the name "current-value". At the bottom, there is an "Editor" section with two input fields: "Regular expression: *" containing `^(\\d\\d\\d)\\s*(\\d\\d\\d)-(\\d\\d\\d\\d)$` and "Replace with:" containing `$1-$2-$3`.

一般運算式 `^(\\d\\d\\d)\\s*(\\d\\d\\d)-(\\d\\d\\d\\d)$` 代表 (nnn) nnn-nnnn，而一般運算式 `$1-$2-$3` 則代表 nnn。此規則會將電話號碼的格式從 (nnn) nnn-nnnn 轉換為 nnn-nnn-nnnn。

3.8.7 Substring

擷取部份字串。

欄位

開始

指定開始字元索引：

- ◆ 索引 0 為第一個字元。
- ◆ 正索引為字串起點的偏移。
- ◆ 索引 -1 為最後一個字元。
- ◆ 負索引為從最後一個字元朝向字元起點的偏移。

例如，如果起點指定為 -2，則會從結尾讀取第一個字元來起算。如果指定 -3，則會從結尾算起 2 個字元處開始。

長度

從起點開始，要包含在子字串中的字元數目。負數會解譯為 (段落總數 + 長度) + 1。例如，-1 即代表整個長度或原始字串。如果指定 -2，則長度為整個 -1。如果字串有 5 個字元，且長度為 -1，則數目 = (5 + (-1)) + 1 = 5, -2 = (5 + (-2)) + 1 = 4，以此類推。

範例

此範例將電子郵件地址設為 name@slartybartfast.com，其中名稱等於「名」的第一個字元加上「姓」。該規則的名稱為「規則：從名和姓建立電子郵件地址」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

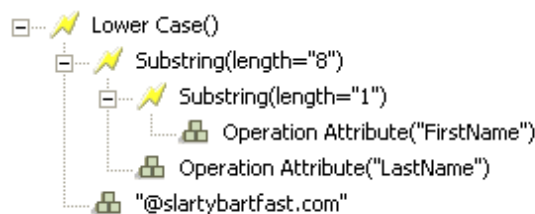
[Push back in email changing](#)

條件

- 如果 類別名稱 等於 "User"
- 且 如果 操作屬性 'Email' 變更

動作

- 設定來源屬性值("Email",目的屬性("Internet EMail Address"))
- 條紋化操作屬性("Email")



Substring 記號在「設定目的屬性值」動作中會使用兩次。該記號會採用 First Name 屬性的第一個字元，並將 Last Name 屬性的八個字元一併新增，以形成一個子字串。

3.8.8 Upper Case

將字串中的字元轉換為大寫。

範例

此範例會將「使用者」物件的名和姓屬性轉換為大寫。規則名稱為「原則：將名/姓轉換為大寫」，可從 Novell 的支援網站下載。如需相關資訊，請參閱「[可下載的 Identity Manager 規則](#)」，第 32 頁。

The screenshot shows a rule configuration window titled "Convert First/Last name to uppercase". It is divided into two main sections: "條件" (Conditions) and "動作" (Actions).
Under "條件", there are three conditions:

- 如果 類別名稱 等於 "User"
- 且
- 如果 操作屬性 'Given Name' 變更
- 或 如果 操作屬性 'Surname' 變更

Under "動作", there are two actions:

- 重新格式化操作屬性("Given Name",大寫(操作屬性("Given Name")))
- 重新格式化操作屬性("Surname",大寫(操作屬性("Surname")))

Upper Case()
Operation Attribute("Given Name")

3.9 值

本節包含常見「規則產生器」值的清單。

3.9.1 比較模式

模式	描述
case	區分大小寫的逐字元比較。
nocase	不區分大小寫的逐字元比較。
regex	整個字串的一般運算式比對。預設為不區分大小寫，但可能會被運算式中的逸出所變更。 請參閱 Sun 的 Java 網站 (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html) 和 Sun 的 Java 網站 (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#matches())。 使用了 CASE_INSENSITIVE、DOTALL 和 UNICODE_CASE 等模式選項，但是可以藉由使用適當的內嵌逸出加以反轉。
src-dn	使用適合來源資料儲存之 DN 格式的語意進行比較。
dest-dn	使用適合目的資料儲存之 DN 格式的語意進行比較。
numeric	以數字的方式進行比較。
octet	比較八進位 (Base64 編碼) 值。

模式	描述
structured	根據屬性的結構化語法比較規則，比較結構化屬性。

Novell® Credential Provisioning Policies for Identity Manager 3 會增強任何 Identity Manager 驅動程式的使用者提供功能，方法是提供同時將應用程式身分證明提供給 Novell SecretStore® 和 Novell SecureLogin 身分證明儲存機制的功能。此外，該產品還會在需要肯定認證功能的環境中提供 SecureLogin 通關密語的問題與回答。

這些功能可增強使用者單次登入 (Single Sign-On, SSO) 體驗，並增加「單次登入」技術的投資報酬，方法是消除 SecureLogin 帳戶資訊的啓始設定，對應用程式身分證明提供額外的安全性，並減少通常與提供給使用者「單次登入」身分證明儲存相關聯的重複工作。此外，「身分證明提供」規則可以使用 Identity Manager 規則自動取消提供應用程式身分證明，以防止存取應用程式資料。

- ◆ 「具有 Novell SecureLogin 的身分證明提供規則」，第 307 頁
- ◆ 「實作具有 SecureLogin 的身分證明提供規則」，第 309 頁
- ◆ 「具有 Novell SecretStore 的身分證明提供規則」，第 328 頁
- ◆ 「實作具有 SecretStore 的身分證明提供規則」，第 330 頁

4.1 具有 Novell SecureLogin 的身分證明提供規則

「身分證明提供」規則可讓您自動提供 SecureLogin 所支援的應用程式身分證明。此主題記載了在 Identity Manager 中設定物件和規則組態時所需的步驟，但不包含任何 SecureLogin 元件的部署和組態資訊。如需 SecureLogin 文件，請參閱 [Novell SecureLogin 6.0 文件 \(http://www.novell.com/documentation/securelogin60/index.html\)](http://www.novell.com/documentation/securelogin60/index.html)。

實作具有 SecureLogin 的「身分證明提供」需要有儲存機制物件、應用程式物件及規則。儲存機制物件和應用程式物件會儲存 SecureLogin 資訊，以便讓 Identity Manager 可以使用。規則用於啓用驅動程式，以使用「身分證明提供」。如需相關資訊，請參閱「實作具有 SecureLogin 的身分證明提供規則」，第 309 頁。

您也可以設定下列選項的組態：

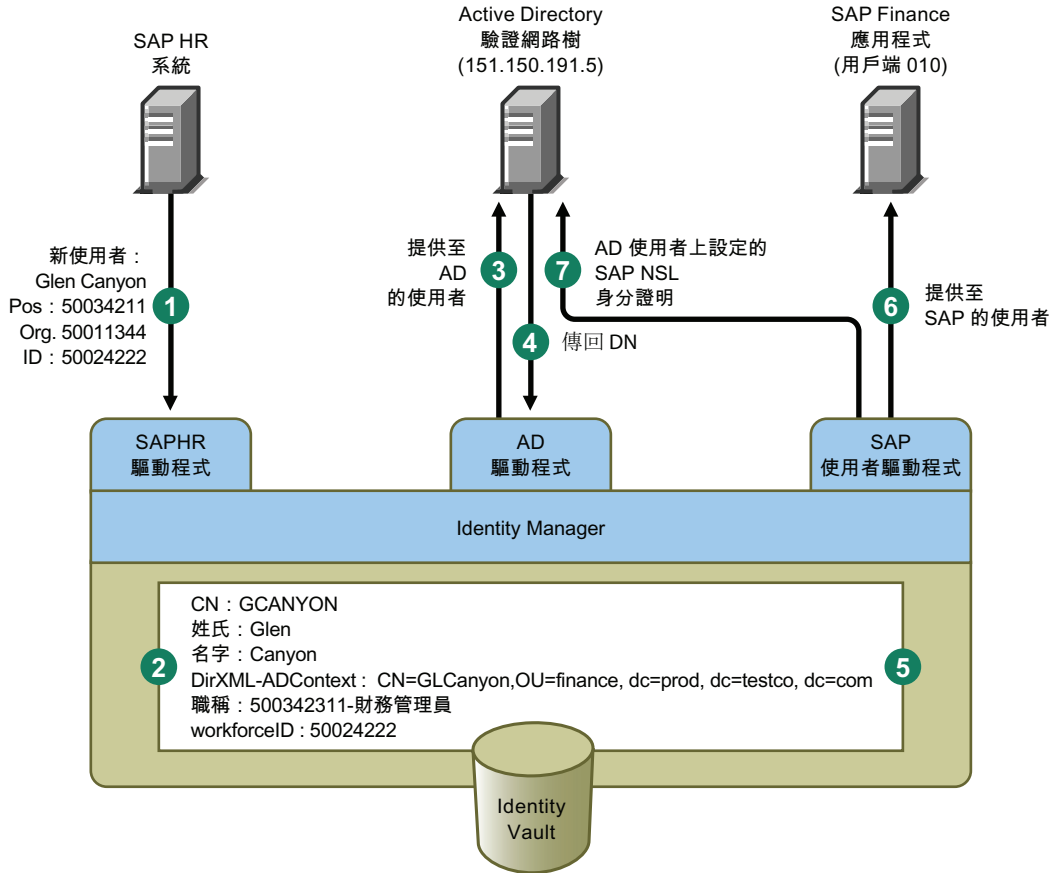
- ◆ 「身分證明提供」可由「發行者」通道、「訂閱者」通道或兩者同時提供。
- ◆ SecureLogin 同步化可做為應用程式密碼同步化的一部份發生，也可以由某個其他事件觸發。
- ◆ 「Web 服務」身分證明可在未提供帳戶給應用程式的情況下提供。
- ◆ 啓始的 SecureLogin 通關密語的問題和回答都可以提供。

特性 4-1 頁上 308 所顯示的是一種典型卻又簡易的案例，其中涉及將 SecureLogin 身分證明提供給財務部門中 SAP® Finance 應用程式的新「使用者」。之所以會使用 SAP 使用者提供，是因為它是一種需要登入參數多於需要提供給大多數應用程式之典型使用者名稱和密碼的應用程式。

此部門透過 SAP HR 系統和 Identity Manager，提供新使用者進入 Identity Vault。視組織資訊而定，「使用者」物件接著會提供到 Active Directory 上實作的部門認證樹中。這是新使用者驗證網路的地方，因此也是 SecureLogin 身分證明儲存機制所使用的位置。在 Identity Manager 後續提供使用者給各種財務應用程式的同時，使用者對於那些系統的身分證明會同步化到 Active Directory 中的 SecureLogin 儲存。

特性 4-1 所顯示的是正在提供使用者 Glen 的驗證身分證明。當 Glen 驗證其部門的 Active Directory 驗證領域並啟動 SecureLogin 用戶端時，他不需要在該系統上輸入（甚至不需要知道）密碼，便能單次登入其 SAP Finance 帳戶。

特性 4-1 具有 *SecureLogin* 的身分證明提供



特性 4-1 說明下列步驟：

1. SAP HR 系統會發行名為 Glen Canyon 之新雇用使用者的資料。Identity Manager SAP HR 驅動程式則會處理此資料。
2. Identity Vault 中會建立新的「使用者」物件，其 CN 值為 GCANYON 且 workforceID 值為 50024222。由於此使用者是指派給其公司的「財務」組織，所以他需要在 finance.prod.testco.com 的領域驗證「財務」部門的 Active Directory 伺服器。同步化該領域的 Identity Manager Active Directory 驅動程式現在會使用 Identity Vault 資訊。
3. Glen 會提供給「財務」部門的 Active Directory 伺服器。
4. 該驅動程式是設定為取得 Glen 的完全可辨識 LDAP 名稱：CN=GLCanyon，OU=finace，dc=prod，dc=testco，dc=com。
5. 驅動程式會在 Identity Vault 中將該名稱放入 GCANYON 使用者的 DirXML-ADContext 屬性。

既然 Identity Vault 中有了必要的屬性，「SAP 使用者管理」驅動程式便會處理 GCANYON 物件的屬性。

6. 因為 Glen 位於「財務」組織，所以驅動程式會在 SAP Finance 伺服器上提供 SAP 使用者帳戶 GCANYON。

7. 帳戶建立成功後，「SAP 使用者管理」驅動程式規則會將 Glen 的 SAP 驗證身分證明提供給他的 AD 使用者帳戶。因為該指令為「新增」操作，所以規則也會提供他的 SecureLogin 通關密語的問題和回答。

4.2 實作具有 SecureLogin 的身分證明提供規則

可以自定具有 SecureLogin 之「身分證明提供」規則實作的幅度極大。實作的步驟會視安裝 SecureLogin 的平台、進行提供的應用程式和涉及的 Identity Manager 驅動程式為何而有所不同。

若要實作具有 SecureLogin 的「身分證明提供」規則，請參閱下列主題：

- ◆ 「符合具有 Novell SecureLogin 之身分證明提供規則的要求」，第 309 頁
- ◆ 「延伸 Novell SecureLogin 的 LDAP 綱要」，第 309 頁
- ◆ 「決定 Novell SecureLogin 的部署組態參數」，第 310 頁
- ◆ 「建立 Novell SecureLogin 的儲存機制物件」，第 312 頁
- ◆ 「建立 Novell SecureLogin 的應用程式物件」，第 319 頁
- ◆ 「設定 Novell SecureLogin 的身分證明提供規則組態」，第 325 頁

4.2.1 符合具有 Novell SecureLogin 之身分證明提供規則的要求

若要使用具有 SecureLogin 的「身分證明提供」規則，必須備妥下列項目：

- ◆ 含支援套件 1 的 Identity Manager 3.0
 - ◆ 必須安裝於 eDirectory™ 8.7x；不支援 eDirectory 8.8。
 - ◆ 驗證 jso.jar、idmcp.jar 和 jnet.jar 是否位於 Identity Manager Java 程式庫的標準位置。
- ◆ Novell SecureLogin 6.0 或以上版本

在驗證環境符合要求後，請繼續進行「[延伸 Novell SecureLogin 的 LDAP 綱要](#)」，第 309 頁。

4.2.2 延伸 Novell SecureLogin 的 LDAP 綱要

在將 SecureLogin 部署於 eDirectory 伺服器後，會使用名為 ndsschema.exe 的工具，以一組 SecureLogin 屬性延伸 eDirectory 綱要，且這些屬性用於儲存「使用者」和容器物件上加密的身分證明、規則等。這些屬性為：

- ◆ Prot:SSO Auth
- ◆ Prot:SSO Entry
- ◆ Prot:SSO Entry Checksum
- ◆ Prot:SSO Profile
- ◆ Prot:SSO Security Prefs
- ◆ Prot:SSO Security Prefs Checksum

這些是 eDirectory 的特定屬性，也是讓 SecureLogin 產品能正常運作的必要屬性。Identity Manager 3.0 支援套件 1 所提供的提供應用程式介面 (Application Programming Interface, API)

使用 LDAP 名稱空間執行其功能，如此便可與 SecureLogin 身分證明儲存一起使用。若要提供 LDAP 映射給上面所列的屬性，必須使用隨附於 SecureLogin 產品的第二項工具。該工具名稱爲 ldapschema.exe 並用於 eDirectory 環境，可提供 LDAP 名稱空間映射給 eDirectory 屬性。

執行 ldapschema.exe 後，請檢查 iManager 中的 LDAP Group 屬性，並藉此驗證映射。

- 1 在 iManager 中，按一下「LDAP > LDAP 選項」。
- 2 選取與您裝載 SecureLogin 之 eDirectory 伺服器相關聯的 LDAP Group。
- 3 從 LDAP Group 內容頁面中選取「屬性映射」選項，並且驗證上述屬性是否映射至下列的「LDAP 主要屬性」：
 - ◆ protocom-SSO-Auth-Data
 - ◆ protocom-SSO-Entries
 - ◆ protocom-SSO-Entries-Checksum
 - ◆ protocom-SSO-Profile
 - ◆ protocom-SSO-Security-Prefs
 - ◆ protocom-SSO-Security-Prefs-Checksum

綱要延伸後，請繼續進行「決定 Novell SecureLogin 的部署組態參數」，第 310 頁。

4.2.3 決定 Novell SecureLogin 的部署組態參數

爲了提供特性 4-1 所示的部署案例中描述的同步化功能，第一步就是蒐集與 Identity Manager 和 SecureLogin 環境相關的所有企業流程資訊。您可以列印表格 4-1，「SecureLogin 的身分證明提供規則工作表」，第 310 頁，並用它做爲記錄資訊的工作表。

表格 4-1 SecureLogin 的身分證明提供規則工作表

需要組態資訊	資訊
1) 要設定哪些應用程式的組態，以供 SecureLogin 的「單一登入」提供使用？	
2) 驗證 SecureLogin 應用程式定義的組態是否已在驗證伺服器上預先設定，且可由提供給那些系統的新使用者所承襲。	
3) SecureLogin 儲存機制伺服器的 DNS 名稱或 IP 位址。	
4) SecureLogin 儲存機制伺服器的 SSL LDAP 連接埠。	
5) SecureLogin 儲存機制伺服器之管理員的完全合法 LDAP 可辨識名稱。	
6) SecureLogin 儲存機制伺服器之管理員的密碼。	
7) 從 SecureLogin 伺服器輸出之 SSL 證書的完整路徑和名稱。證書必須要在 Identity Manager 伺服器本地上。	

需要組態資訊	資訊
8) 決定是多個驅動程式會使用一個 SecureLogin 儲存機制，還是每個驅動程式都會使用個別的儲存機制。	
9) 每個 SecureLogin 應用程式的應用程式 ID。	
10) 尋找每個應用程式的所有必要驗證金鑰。例如「使用者名稱」、「密碼」、「用戶端」，以及「語言」。這些對每個應用程式而言可能都不同。	
11) 決定任何一個驗證金鑰值是否可與靜態值一併設定。	
12) 針對對於每個使用者而言不同或可以不同的非靜態值，記下非靜態資訊 (事件資訊或 Identity Vault 屬性值) 的來源。	
13) 如果您正在驅動程式上實作 SecureLogin 提供，且驅動程式也正在將密碼同步化至目標應用程式，請決定 SecureLogin 提供要在密碼於目標應用程式伺服器中設定之前或之後發生。	
14) 要在其中儲存儲存機制和應用程式物件之「驅動程式」物件的名稱。(可為不同的驅動程式。)	
15) 決定目標應用程式的「使用者」物件 DN。	
16) 如果您在實作 SecureLogin 通關密語，請決定通關密語的問題和回答。	問題：回答：

範例提供組態資料

下列範例資料使用提供案例，提供某位使用者的 SecureLogin 身分證明給 SAP Finance 伺服器，以便用於 Finance Active Directory 驗證樹中的使用者。

表格 4-2 SecureLogin 的範例身分證明提供規則工作表

需要組態資訊	資訊
1) 要設定哪些應用程式的組態，以供 SecureLogin 的「單一登入」提供使用？	SAP Finance 應用程式
2) 驗證 SecureLogin 應用程式定義的組態是否已在驗證伺服器上預先設定，且可由提供給那些系統的新使用者所承襲。	已驗證
3) SecureLogin 儲存機制伺服器的 DNS 名稱或 IP 位址。	151.150.191.5
4) SecureLogin 儲存機制伺服器的 SSL LDAP 連接埠。	636
5) SecureLogin 儲存機制伺服器之管理員的完全合法 LDAP 可辨識名稱。	cn=admin, ou=prod, dc=testco, dc=.com
6) SecureLogin 儲存機制伺服器之管理員的密碼。	dixml

需要組態資訊	資訊
7) 從 SecureLogin 伺服器輸出之 SSL 證書的完整路徑和名稱。證書必須要在 Identity Manager 伺服器本地。	c:\novell\nds\FinanceAD.cer
8) 決定是多個驅動程式會使用一個 SecureLogin 儲存機制，還是每個驅動程式都會使用個別的儲存機制。	此例中只有一個儲存機制。
9) 每個 SecureLogin 應用程式的應用程式 ID。	SAP - 151.150.191.27
10) 尋找每個應用程式的所有必要驗證金鑰。例如「使用者名稱」、「密碼」、「用戶端」，以及「語言」。這些對每個應用程式而言可能都不同。	SAP Client 010 Login Parameter Client SAP Client 010 Login Parameter Language SAP Client 010 Login Parameter Username SAP Client 010 Login Parameter Password
11) 決定任何一個驗證金鑰值是否可與靜態值一併設定。	SAP Client 010 Login Parameter Client:"010" SAP Client 010 Login Parameter Language: "EN"
12) 針對對於每個使用者而言不同或可以不同的非靜態值，記下非靜態資訊 (事件資訊或 Identity Vault 屬性值) 的來源。	SAP Client 010 登入參數使用者名稱: Identity Vault 屬性 "sapUsername" 的 SAP Client 010 登入參數密碼: 事件 <password>
13) 如果您正在驅動程式上實作 SecureLogin，且驅動程式也正在將密碼同步化至目標應用程式，請決定 SecureLogin 提供要在密碼於目標應用程式伺服器中設定之前或之後發生。	之後
14) 要在其中儲存儲存機制和應用程式物件之「驅動程式」物件的名稱。(可為不同的驅動程式。)	SAP 驅動程式
15) 決定目標應用程式的「使用者」物件 DN。	Identity Vault 屬性 "DirXML-ADContext"
16) 如果您即將提供 SecureLogin 通關密語，請決定通關密語的問題和回答。	問題: "員工碼?" 回答: Identity Vault 屬性 "workforceID"

其他環境資訊：

- ◆ 「財務」部門 AD 樹會做為所有 Finance 應用程式的 SecureLogin 儲存機制使用。
- ◆ 所有的財務部門提供驅動程式都位於稱為 Finance Drivers 的驅動程式集中。
- ◆ 當 Identity Vault 屬性 employeeStatus 設為值 "I" 時，必須刪除 SAP 使用者帳戶，且必須從 Active Directory 使用者中移除該 SAP 使用者帳戶的 SecureLogin 身分證明。

所有的組態資料都決定好後，請繼續進行「[建立 Novell SecureLogin 的儲存機制物件](#)」，第 312 頁。

4.2.4 建立 Novell SecureLogin 的儲存機制物件

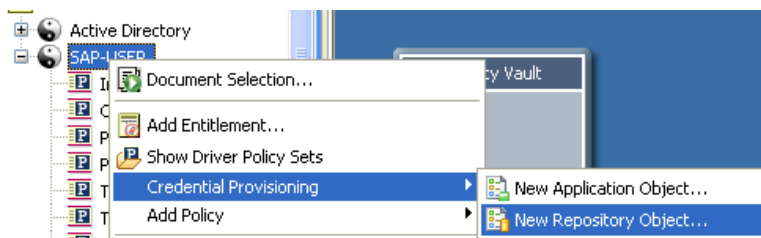
儲存機制物件會儲存 SecureLogin 的靜態組態資訊。儲存機制資訊獨立於使用應用程式身分證明的應用程式以外。不論已連接系統為何 (如 SAP、PeopleSoft*、Notes* 等)，此資訊均適用於所有的提供事件。儲存機制物件可在 Designer 或 iManager 中建立。

- ◆ 「[在 Designer 中建立 Novell SecureLogin 的儲存機制物件](#)」，第 313 頁
- ◆ 「[在 iManager 中建立 Novell SecureLogin 的儲存機制物件](#)」，第 316 頁

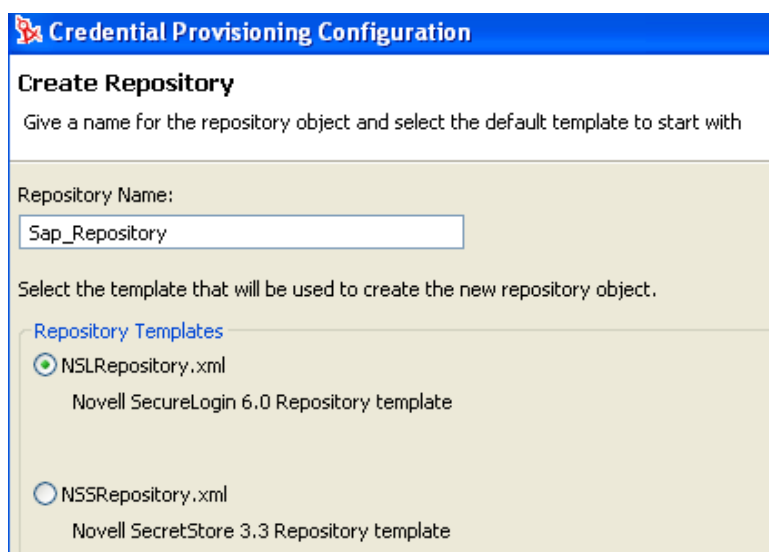
在 **Designer** 中建立 **Novell SecureLogin** 的儲存機制物件

下列是您可以用來在 **Designer** 中建立儲存機制物件的眾多方法之一。

- 1 在大綱檢視窗中，以滑鼠右鍵按一下儲存機制物件要儲存的所在驅動程式物件。
- 2 按一下「身分證明提供 > 新儲存機制物件」。

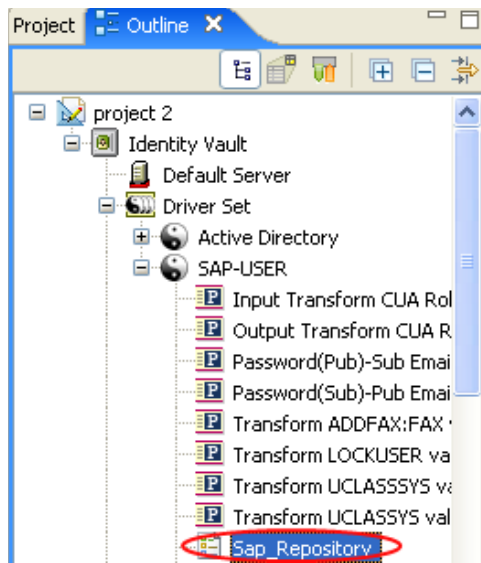


- 3 指定儲存機制物件的名稱。
- 4 選取「*NSLRepository.xml*」以使用 **SecureLogin** 範本。

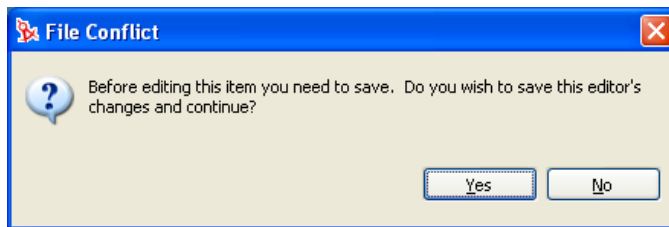


- 5 按一下「確定」。

- 6 在大綱檢視窗中連接兩下儲存機制物件，新增組態資訊。



- 7 按一下「是」儲存新儲存機制物件。



- 8 指定 SecureLogin 伺服器的 DNS 名稱或 IP 位址。請參閱工作表項目 3)。

SecureLogin Server Name or Address:
151.150.191.5

- 9 指定 SecureLogin 伺服器的 SSL 連接埠。請參閱工作表項目 4)。

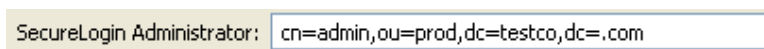
SecureLogin Server SSL Port: 636 ⓘ

- 10 指定從 SecureLogin 伺服器輸出之 SSL 證書的完整路徑。路徑必須包含證書名稱，且必須要在 Identity Manager 伺服器本地。請參閱工作表項目 7)。

SecureLogin Server SSL Certificate Path:
c:\novell\nds\FinanceAD.cer

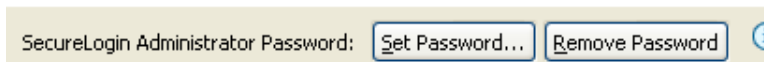
SecureLogin 伺服器可在多種類型的平台上執行。如需如何輸出 SSL 證書的相關資訊，請參閱平台特定的文件。

11 指定 SecureLogin 管理員的完全合法 LDAP 可辨識名稱。請參閱工作表項目 5)。




SecureLogin Administrator: cn=admin,ou=prod,dc=testco,dc=.com

12 按一下「設定密碼」。





SecureLogin Administrator Password: Set Password... Remove Password

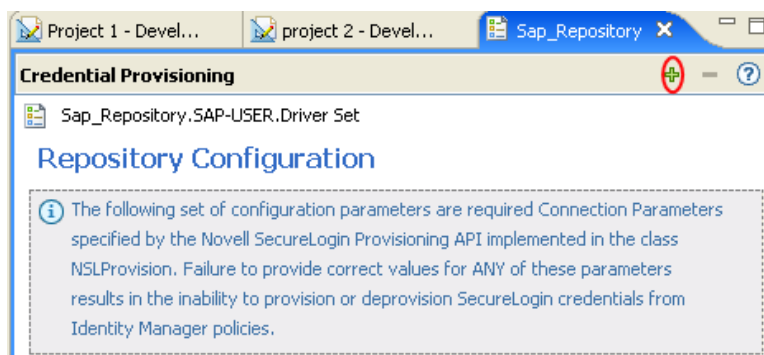
13 指定兩次 SecureLogin 管理員的密碼，然後按一下「確定」。請參閱工作表項目 6)。



Change Password dialog box with fields for 'Enter password:' and 'Re-enter password:' and buttons for 'OK' and 'Cancel'.

14 檢視資訊，然後按一下「儲存」圖示  儲存資訊。

15 (選擇性) 如果想要建立儲存機制物件的其他組態參數，請按一下「新增項目」的  圖示。



Credential Provisioning window showing 'Repository Configuration' and a message: 'The following set of configuration parameters are required Connection Parameters specified by the Novell SecureLogin Provisioning API implemented in the class NSLProvision. Failure to provide correct values for ANY of these parameters results in the inability to provision or deprovision SecureLogin credentials from Identity Manager policies.'

15a 指定參數的名稱。

15b 指定參數的顯示名稱。

15c 指定供您參考的參數描述。

參數會儲存為字串。


Name:

Display name:

Description:

Type:
string ▼

15d 按一下「確定」。

15e 按一下「儲存」圖示  儲存儲存機制物件。

儲存機制物件建立後，請繼續進行「[建立 Novell SecureLogin 的應用程式物件](#)」，第 319 頁。

在 iManager 中建立 **Novell SecureLogin** 的儲存機制物件

- 1 在 iManager 中，選取「身分證明提供 > 組態」。
- 2 瀏覽至儲存機制物件儲存的所在「驅動程式」物件並加以選取，然後按一下「確定」。

Select IDM Container

Select the container that holds the Credential Provisioning objects.

IDM Container:

OK Cancel

- 3 按一下「新增」建立儲存機制。

IDM 容器：GroupWise.drivers&t1.novell

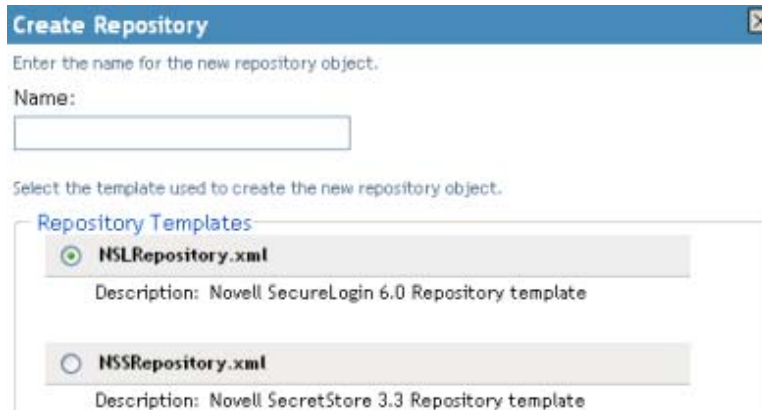
儲存機制 應用程式

新增... | 刪除 | 選取容器...

名稱

找不到任何儲存機制 - 選取「新增」

- 4 指定儲存機制物件的名稱，然後選取「*NSLRepository.xml*」以使用 SecureLogin 範本建立儲存機制。



- 5 按一下「確定」。
- 6 指定 SecureLogin 伺服器的 DNS 名稱或 IP 位址。請參閱工作表項目 3)。

SecureLogin Server Name or Address ⓘ

- 7 指定 SecureLogin 伺服器的 SSL 連接埠。請參閱工作表項目 4)。

SecureLogin Server SSL Port ⓘ

- 8 指定從 SecureLogin 伺服器輸出之 SSL 證書的完整路徑。路徑必須包含證書名稱，且必須要在 Identity Manager 伺服器本地。請參閱工作表項目 7)。

SecureLogin Server SSL Certificate Path ⓘ

SecureLogin 伺服器可在多種類型的平台上執行。如需如何輸出 SSL 證書的步驟，請參閱平台特定的文件。

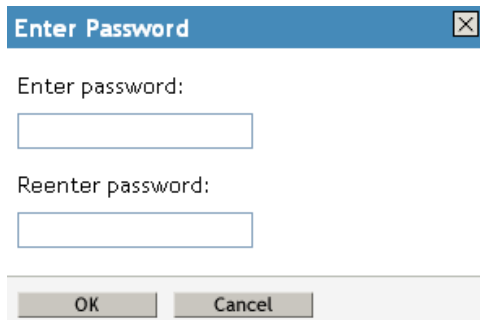
- 9 指定 SecureLogin 管理員的完全合法 LDAP 可辨識名稱。請參閱工作表項目 5)。

SecureLogin Administrator ⓘ

- 10 按一下「設定密碼」。

SecureLogin Administrator Password ⓘ [Set password](#)

- 11 指定兩次 SecureLogin 管理員的密碼，然後按一下「確定」。請參閱工作表項目 6)。



Enter Password

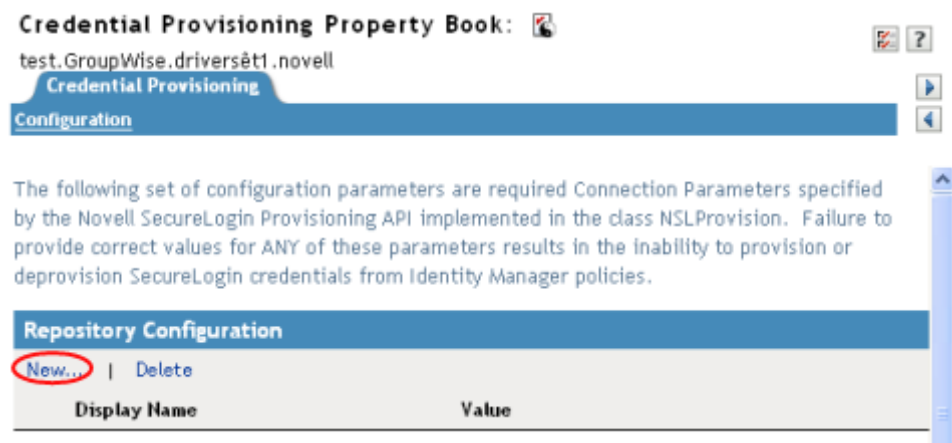
Enter password:

Reenter password:

OK Cancel

- 12 檢視指定的值，然後按一下「確定」。

- 13 (選擇性) 如果需要建立儲存機制物件的其他組態參數，請按一下「新增」。



Credential Provisioning Property Book: test.GroupWise.driverset1.novell

Credential Provisioning

Configuration

The following set of configuration parameters are required Connection Parameters specified by the Novell SecureLogin Provisioning API implemented in the class NSLProvision. Failure to provide correct values for ANY of these parameters results in the inability to provision or deprovision SecureLogin credentials from Identity Manager policies.

Repository Configuration

New... | Delete

Display Name	Value
--------------	-------

- 13a 指定參數的名稱。
- 13b 指定參數的顯示名稱。
- 13c 指定供您參考的參數描述。

參數會儲存為字串。

Global Configuration Value Definition

Global Configuration Values are a means through which the behavior of an Identity Manager driver configuration can be changed without requiring any policy to be changed.

Name:

Display name:

Description:

Type:
string

13d 按一下「確定」。

儲存機制物件建立後，請繼續進行「在 iManager 中建立 Novell SecureLogin 的應用程式物件」，第 322 頁。

4.2.5 建立 Novell SecureLogin 的應用程式物件

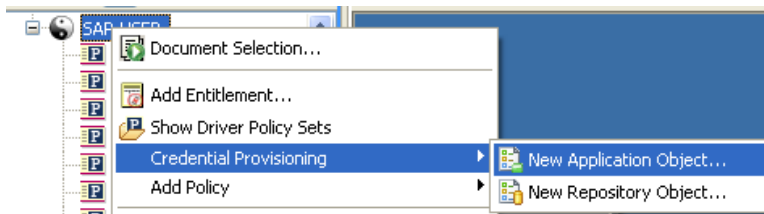
應用程式物件會儲存 SecureLogin 的應用程式驗證參數值。應用程式資訊是使用應用程式身分證明的應用程式特有的 (例如 GroupWise® 用戶端資訊或 SAP 資料庫用戶端資訊)。應用程式物件可在 Designer 或 iManager 中建立。

- ◆ 「在 Designer 中建立 Novell SecureLogin 的應用程式物件」，第 319 頁
- ◆ 「在 iManager 中建立 Novell SecureLogin 的應用程式物件」，第 322 頁

在 Designer 中建立 Novell SecureLogin 的應用程式物件

下列是您可以用來在 Designer 中建立應用程式物件的眾多方法之一。

- 1 在大綱檢視窗中，以滑鼠右鍵按一下應用程式物件要儲存的所在驅動程式物件。
- 2 按一下「身分證明提供 > 新應用程式物件」。

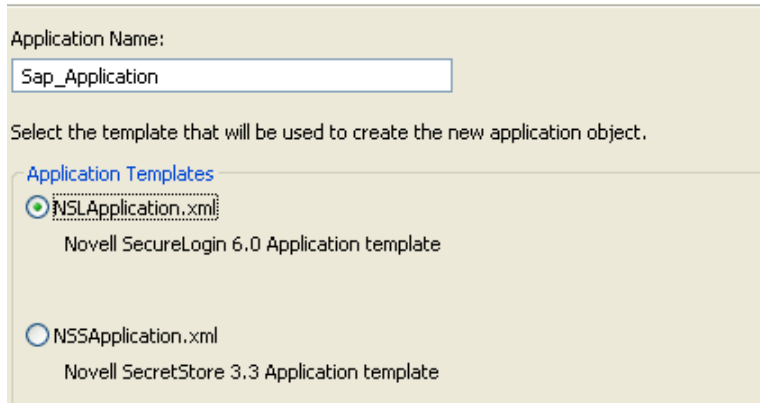


- 3 指定應用程式物件的名稱。

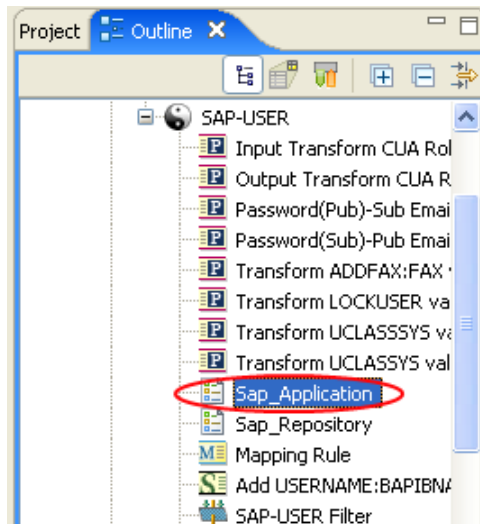
- 4 選取 「*NSLApplication.xml*」 以使用 SecureLogin 範本。

Create Application

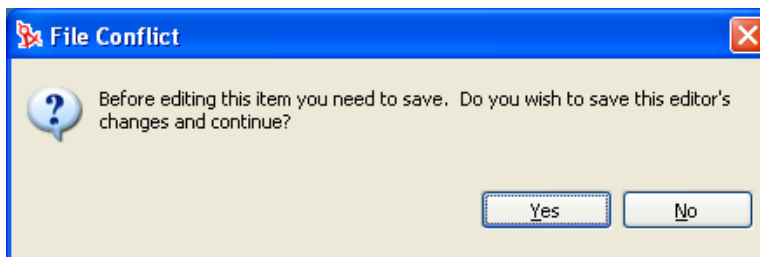
Give a name for the application object and select the default template to start with



- 5 按一下「確定」。
- 6 在大綱檢視窗中連按兩下應用程式物件，新增組態資訊。



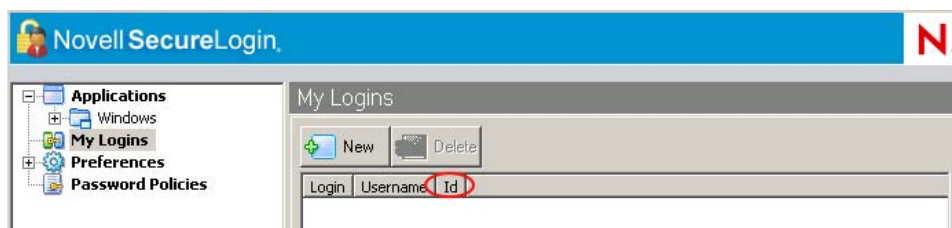
- 7 按一下「是」儲存新應用程式物件。



- 8 指定 SecureLogin 應用程式 ID。請參閱工作表項目 9)。

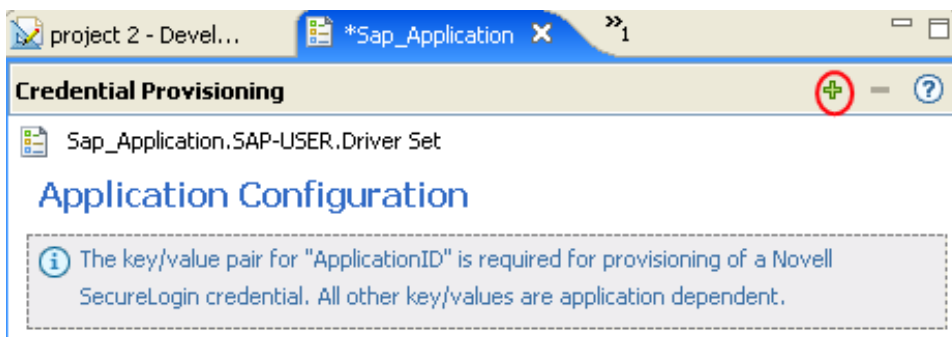
SecureLogin Application ID:

若要尋找 SecureLogin 中的應用程式 ID，請按一下「我的登入」。應用程式 ID 儲存於「ID」欄位中。



- 9 按一下「儲存」圖示  儲存應用程式。

- 10 按一下「新增項目」圖示  新增應用程式需要的驗證金鑰。



- 10a 指定驗證金鑰的名稱。

- 10b 指定驗證金鑰的顯示名稱。


- 10c 指定供您參考的驗證金鑰描述。

驗證金鑰會儲存為字串。

Name:

Display name:

Description:

Type:
string 

10d 按一下「確定」。

10e 針對每個需要輸入的新驗證金鑰，重複**步驟 10**。

若要尋找應用程式的驗證金鑰，請在應用程式中手動建立使用者的 SecureLogin 身分證明，並讓該使用者登入。使用者登入後，驗證金鑰資訊便會顯示於 SecureLogin 管理視窗中的「我的登入」下。

11 如果驗證金鑰是所有使用者身分證明所共用的靜態值，則指定驗證金鑰。

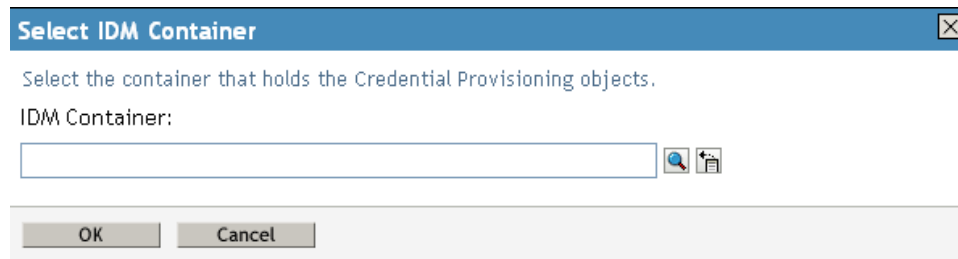
12 按一下「儲存」圖示  儲存應用程式。

應用程式物件建立後，請繼續進行「設定 Novell SecureLogin 的身分證明提供規則組態」，[第 325 頁](#)。

在 iManager 中建立 Novell SecureLogin 的應用程式物件

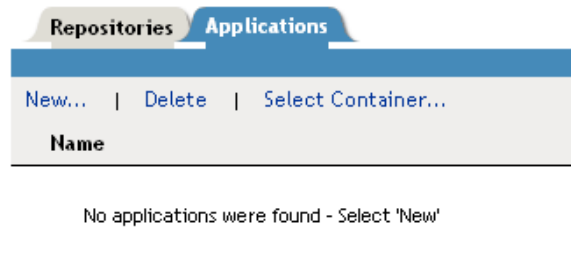
1 在 iManager 中，選取「身分證明提供 > 組態」。

2 瀏覽至應用程式物件儲存的所在「驅動程式」物件並加以選取，然後按一下「確定」。



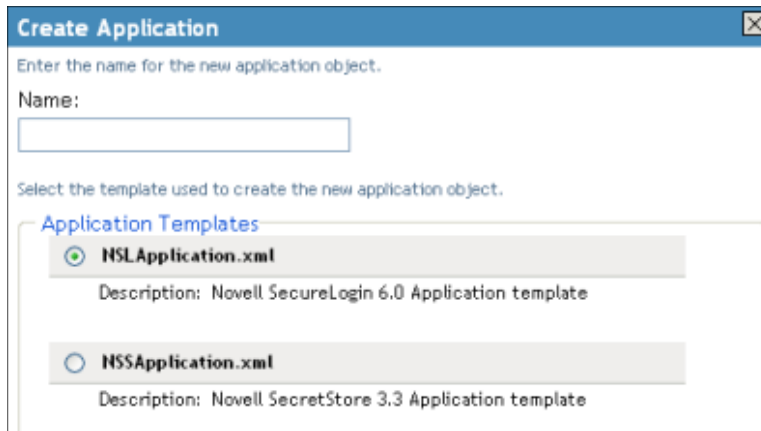
3 選取「應用程式」索引標籤，然後按一下「新增」。

Container: Delimited Text.DriverSet.Novell



4 指定應用程式物件的名稱。

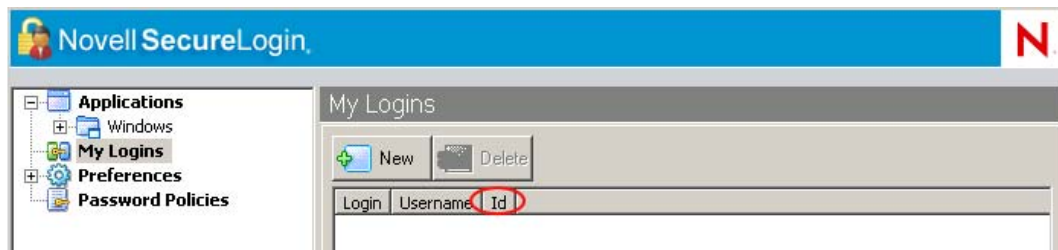
- 5 選取「NSLApplication.xml」以使用 SecureLogin 範本建立應用程式。



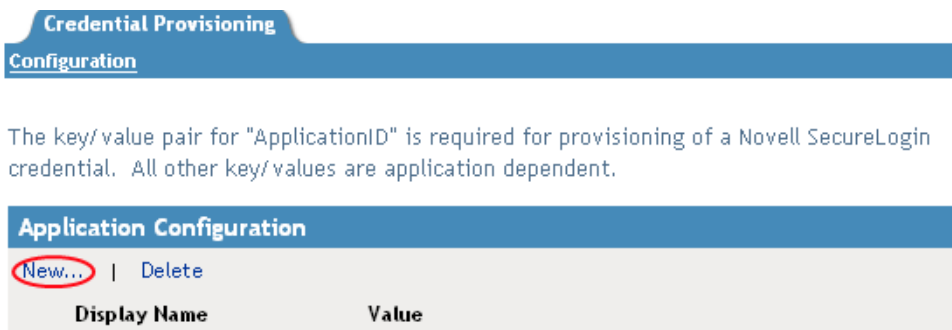
- 6 按一下「確定」。
- 7 指定「SecureLogin 應用程式 ID」。請參閱項目工作表 9)。

SecureLogin Application ID ⓘ

若要尋找 SecureLogin 中的應用程式 ID，請按一下「我的登入」。應用程式 ID 儲存於「ID」欄位中。



- 8 按一下「新增」建立驗證金鑰參數。請參閱工作表項目 10)。



- 8a 指定驗證金鑰的名稱。
- 8b 指定驗證金鑰的顯示名稱。
- 8c 指定供您參考的驗證金鑰描述。

驗證金鑰會儲存為字串。

Global Configuration Value Definition

Global Configuration Values are a means through which the behavior of an Identity Manager driver configuration can be changed without requiring any policy to be changed.

Name:

Display name:

Description:

Type:
string

若要尋找應用程式的驗證金鑰，請在應用程式中手動建立使用者的 SecureLogin 身分證明，並讓該使用者登入。使用者登入後，驗證金鑰資訊便會顯示於 SecureLogin 管理視窗中的「我的登入」下。

- 8d 按一下「確定」。
- 8e 如果驗證金鑰值為靜態值，則指定該值，然後按一下「確定」。

Application Configuration	
New... Delete	
Display Name	Value
<input type="checkbox"/> SecureLogin Application ID ⓘ	<input type="text" value="SAP - 151.150.191.27"/>
<input type="checkbox"/> Client ⓘ	<input type="text" value="010"/>
<input type="checkbox"/> Language ⓘ	<input type="text" value="EN"/>
<input type="checkbox"/> Username ⓘ	<input type="text"/>
<input type="checkbox"/> Password ⓘ	<input type="text"/>

應用程式物件建立後，請繼續進行「設定 Novell SecureLogin 的身分證明提供規則組態」，第 325 頁。

4.2.6 設定 Novell SecureLogin 的身分證明提供規則組態

儲存機制物件和應用程式物件建立後，便需要建立規則以提供 SecureLogin 資訊。規則會使用儲存於儲存機制和應用程式物件中的資訊。「規則產生器」中有三種動作，可以提供 SecureLogin 身分證明：

- ◆ 「清除 SSO 身分證明」，第 325 頁
- ◆ 「設定 SSO 身分證明」，第 326 頁
- ◆ 「設定 SSO 通關密語」，第 326 頁

清除 SSO 身分證明

「清除 SSO 身分證明」動作可讓您清除 SSO 身分證明，所以便可取消提供物件。

特性 4-2 清除 SSO 身分證明

The screenshot shows the 'Action List' configuration interface. At the top, the action is set to 'clear SSO credential'. Below this, there are several input fields and options:

- 'Enter credential store object DN:*' with a search icon.
- A checked checkbox labeled 'Render browsed DN relative to policy'.
- 'Enter target user DN:*' with a search icon.
- A blue link labeled 'Populate the following from an application object'.
- 'Enter application credential ID:*' with a search icon.
- 'Enter login parameter strings:' with a search icon.

- ◆ 輸入身分證明儲存物件 **DN**：瀏覽至儲存機制物件並加以選取。
- ◆ 輸入目標使用者 **DN**：使用「引數產生器」建立目標使用者 DN。請參閱工作表項目 15)。
- ◆ 輸入應用程式身分證明 **ID**：指定應用程式 ID。請參閱工作表項目 9)。
- ◆ 輸入登入參數字串：啟動「字串產生器」並輸入應用程式的每個驗證金鑰。請參閱工作表項目 10)。

設定 SSO 身分證明

「設定 SSO 身分證明」動作可讓您在使用者物件建立或密碼經過修改時，設定 SSO 身分證明。

特性 4-3 設定 SSO 身分證明

The screenshot shows the 'Action List' interface for the 'set SSO credential' action. It includes a dropdown menu with 'Do' selected, and several input fields: 'Enter credential store object DN:*', 'Enter target user DN:*', 'Enter application credential ID:*', and 'Enter login parameter strings:'. There is also a checkbox for 'Render browsed DN relative to policy' which is checked. A link 'Populate the following from an application object' is visible below the 'Enter target user DN:*' field.

- ◆ 輸入身分證明儲存物件 **DN**：瀏覽至儲存機制物件並加以選取。
- ◆ 輸入目標使用者 **DN**：使用「引數產生器」建立目標使用者 **DN**。請參閱工作表項目 15)。
- ◆ 輸入應用程式身分證明 **ID**：指定應用程式 **ID**。請參閱工作表項目 9)。
- ◆ 輸入登入參數字串：啟動「字串產生器」並輸入應用程式的每個驗證金鑰。請參閱工作表項目 10)。

設定 SSO 通關密語

「設定 SSO 通關密語」動作可讓您建立在提供使用者物件時的 SecureLogin 通關密語和回答。

特性 4-4 設定 SSO 通關密語

The screenshot shows the 'Action List' interface for the 'set SSO passphrase' action. It includes a dropdown menu with 'Do' selected, and several input fields: 'Enter credential store object DN:*', 'Enter target user DN:*', and 'Enter question and answer strings:*'. There is also a checkbox for 'Render browsed DN relative to policy' which is checked.

- ◆ 輸入身分證明儲存物件 **DN**：瀏覽至儲存機制物件並加以選取。
- ◆ 輸入目標使用者 **DN**：使用「引數產生器」建立目標使用者 **DN**。請參閱工作表項目 15)。
- ◆ 輸入問題與回答字串：啟動「字串產生器」並輸入通關密語的問題和回答。請參閱工作表項目 16)。

範例身分證明提供規則

您可以實作並自定提供規則，以符合環境需求。下列範例說明如何實作特性 4-1 頁上 308 中呈現之案例的規則。

在「財務」案例中，SecureLogin 提供會發生在 SAP 中的密碼成功設定以後。大多數的必要參數組態都是以靜態方式設定，而且都可以透過儲存機制物件和應用程式物件，供所有的規則使用。不過，有些資料參數並非靜態 (sapUsername、password、DirXML-ADContext 及 workforceID)，而且只有在「SAP 使用者管理」驅動程式 <add> 或 <modify-password> 指令完成，且「SAP 使用者管理」驅動程式 Shim 傳回 <output> 狀態文件後才可使用。<output> 不再包含任何「訂閱者」通道操作屬性，且指令的使用者網路位置也遺失，因此阻止了對於物件的查詢。因此有必要進行下列幾項操作：

- ◆ 確定「SAP 使用者」驅動程式的「訂閱者建立」規則強制非靜態資料參數存在。
- ◆ 將「訂閱者」指令發出至「SAP 使用者」驅動程式 Shim 之前，先快取提供操作所需的非靜態參數。
- ◆ 指令順利完成後取回已快取的資料，以便用於 SecureLogin 提供。

附註：Identity Manager 3.0 支援套件 1 的媒體上有 XML 格式的範例規則。檔名為 SampleInputTransform.xml、SampleSubCommandTransform.xml 及 SampleSubEventTransform.xml。這些檔案位於下列目錄中（視平台而定）：

- ◆ linux\setup\utilities\cred_prov
- ◆ nt\dirxml\utilities\cred_prov
- ◆ nw\dirxml\utilities\cred_prov

如果在安裝公用程式時選取「身分證明提供範例規則」，這些檔案便會安裝到 Identity Manager 伺服器。根據您的平台而定，範例規則會安裝到下列位置：

- ◆ Windows：C:\Novell\NDS\DirXMLUtilities (此為預設值；使用者於安裝時可加以變更)
- ◆ NetWare®：SYS:\System\DirXmlUtilities
- ◆ Linux (eDir 8.7)：/usr/lib/dirxml/rules/credprov

範例規則提供您一個起點，讓您開發適用於個人環境的規則。

操作資料快取

可供所需操作資料之快取使用的機制為 <operation-data> 元素。因為您可能需要從 <add> 或 <modify-password> 指令提供 SecureLogin 帳戶，所以實作非靜態資料快取規則的合理位置就是在「訂閱者指令轉換」規則中。下列範例顯示典型 SecureLogin 提供的 <operation-data> 元素：

```
<operation-data> <nsl-sync-data> <nsl-target-user-dn>
cn=GLCANYON,ou=finance,dc=prod,dc=testco,dc=com </nsl-target-user-dn> <nsl-app-
username>GCANYON</nsl-app-username> <password><!-- content suppressed --></password>
<nsl-passphrase-answer>50024222</nsl-passphrase-answer> </nsl-sync-data> </operation-data>
```

在來自於特性 4-1 頁上 308 的範例「財務」部門案例中，若要填入操作資料封包內容，則需要下列這些值：

- ◆ <nsl-target-user-dn> 元素會從 Identity Vault 中，以 DirXML-ADContext 屬性值加以填入，該值則已由 Active Directory 驅動程式設定。若要確保 AD 驅動程式設定該值時，「SAP 使用者」驅動程式會收到通知，請務必將 DirXML-ADContext 做為通知屬性新增至「訂閱者」過濾器。
- ◆ <nsl-app-username> 會由 sapUsername 屬性值填入，該值對於 <add> 指令而言則會經由「SAP 使用者」驅動程式的「建立」規則來產生，因此可做為操作屬性使用。有了

「SAP 使用者」驅動程式，「SAP 使用者」的名稱值便為關聯值的一部分。這表示就密碼修改事件而言，會從關聯解析名稱。

- ◆ password 元素會由 <add> 或 <modify-password> 指令中的 <password> 元素值填入。
- ◆ <nsi-passphrase-answer> 會由 Identity Vault 中的 workforceID 屬性值填入，該值則已由 SAP HR 驅動程式設定。雖然此值在一開始對 Identity Vault 提供時就要設定，但將 workforceID 做為通知屬性新增至「訂閱者」過濾器仍是一種很好的作法。

SecureLogin 提供

在此案例中，可將操作資料從其中取回，並用於 SecureLogin 身分證明提供的第一個可用位置就是在驅動程式的「輸出轉換」規則中。範例案例中實作了三種規則：

- ◆ 密碼同步化成功後，設定「SecureLogin 身分證明」。
- ◆ 設定 SecureLogin 通關密語和回答
- ◆ 如果應用程式已刪除 (Identity Vault 物件未刪除)，則移除「SecureLogin 身分證明」

附註：SampleInputTransform.xml 檔案中有範例規則，可在密碼同步化成功後設定 SecureLogin 身分證明。該檔案位於 Identity Manager 3.0 支援套件 1 之媒體上的「身分證明提供」資料夾中。

「設定 SecureLogin 身分證明」規則需要確定只有在傳回的指令狀態為 success，且之前設定的 <operation-data> 存在時，提供才會發生。

SecureLogin 取消提供

有許多案例可以使用規則，其中已連接應用程式的使用者帳戶已刪除，而 Identity Vault 帳戶則予以保留。在「財務」案例中，當「使用者」的 Identity Vault 屬性值 employeeStatus 設為 "I" 時，則有必要刪除「SAP 使用者」帳戶並取消提供 SecureLogin 身分證明。為了處理此情況，「SAP 使用者」驅動程式的「訂閱者事件轉換」包含要將修改屬性值轉換為物件刪除的規則。因為刪除指令完成後仍需要 Active Directory 帳戶名稱，所以 <operation-data> 事件需要在 <delete> 指令進行時設定，如此便可在「輸入轉換」規則中供 SecureLogin 取消提供規則使用。

```
<operation-data> <nsi-sync-data> <nsi-target-user-dn>cn=GLCANYON, ou=finance, dc=prod, dc=testco, dc=com </nsi-target-user-dn> </nsi-sync-data> </operation-data>
```

SampleSubEventTransform.xml 檔案的範例「身分證明提供」規則中用於將 <modify> 事件轉換為 <delete> 並建立此元素的規則。

4.3 具有 Novell SecretStore 的身分證明提供規則

「身分證明提供」規則可讓您將應用程式身分證明提供給 Novell SecretStore 儲存機制中的「使用者」物件。提供「使用者伺服器」和「使用者」身分證明的功能是標準 Identity Manager 提供案例的一部分，讓使用者能夠使用更安全、更同步化的 Web「單一登入」。

此文件包含 Identity Manager 中設定物件和規則組態時所需的步驟，但不包含任何 SecretStore 元件的部署和組態資訊。如需 SecretStore 文件，請參閱 [Novell SecretStore 3.3.3 文件 \(http://www.novell.com/documentation/secretstore33/index.html\)](http://www.novell.com/documentation/secretstore33/index.html)。

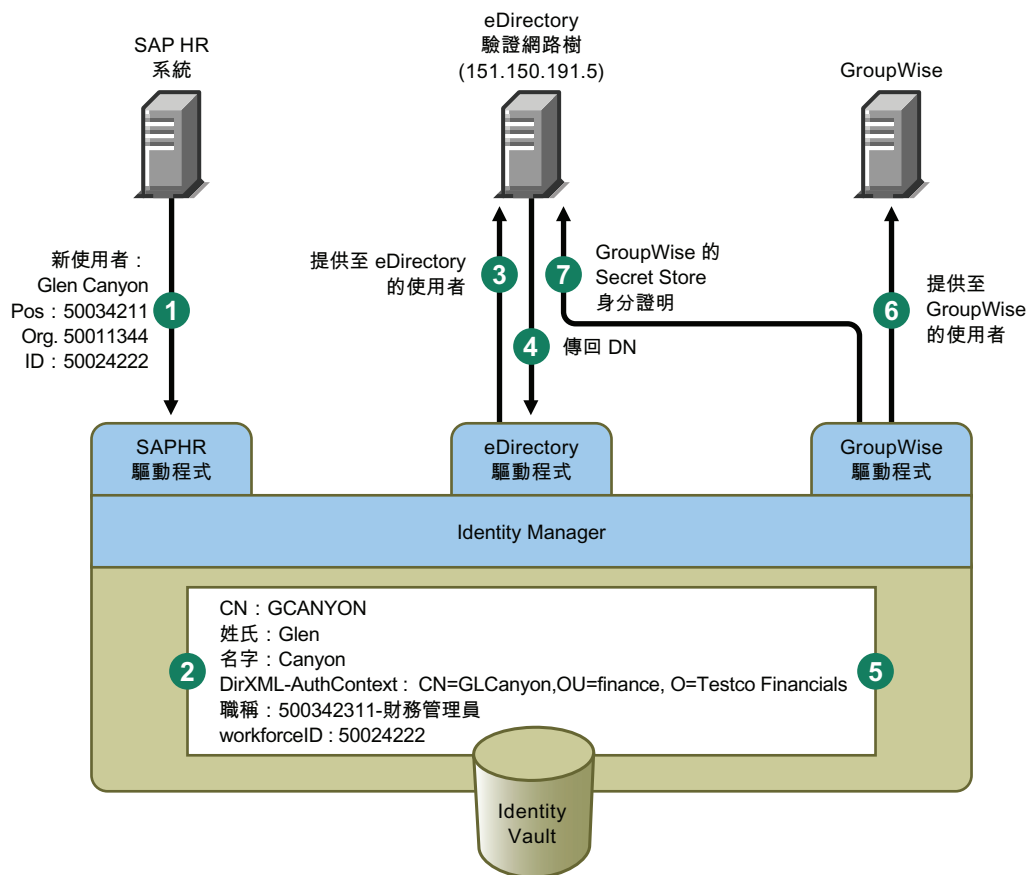
實作具有 SecretStore 的「身分證明提供」需要有儲存機制物件、應用程式物件及建立規則。儲存機制和應用程式物件會儲存 SecretStore 資訊，以便讓 Identity Manager 可以使用。

使用規則後，任何驅動程式都可以啟動，以便使用「身分證明提供」。您也可以設定下列選項的組態：

- ◆ 「身分證明提供」可由「發行者」通道、「訂閱者」通道或兩者同時提供。
- ◆ SecretStore 同步化可做為應用程式密碼同步化的一部份發生，也可以由某個其他事件觸發。
- ◆ 「Web 服務」身分證明可在未提供帳戶給應用程式的情況下提供。

特性 4-1 頁上 308 所顯示的是一種典型卻又簡易的案例，其中涉及將「單一登入」身分證明提供給 GroupWise® 的新使用者。此部門透過 SAP HR 系統和 Identity Manager，提供新使用者進入 Identity Vault。視組織資訊而定，使用者接著會提供到 eDirectory 上實作的部門認證樹中。這是使用者驗證網路的地方，也是 GroupWise 安全性身分證明的儲存機制；Novell iChain® 或 Access Manager® 會從公司防火牆外，使用該儲存機制提供安全「單一登入」功能。在 Identity Manager 後續提供使用者給 GroupWise 的同時，那些系統的身分證明會同步化到驗證樹中的 SecretStore 屬性。

特性 4-5 SecretStore 同步化可做為應用程式密碼同步化的一部份發生，也可以由某個其他事件觸發。



特性 4-5 說明下列提供步驟：

1. SAP HR 系統會發行名為 Glen Canyon 之新雇用使用者的資料。Identity Manager SAP HR 驅動程式則會處理此資料。
2. Identity Vault 中會建立新的「使用者」物件，其 CN 值為 GCANYON 且 workforceID 值為 50024222。由於此使用者是指派給其公司的「財務」組織，所以他需要驗證「財

務」部門的 eDirectory 伺服器。這時同步化該領域的 Identity Manager eDirectory 驅動程式會使用 Identity Vault 資訊。

3. Glen 會提供給「財務」部門的 eDirectory 伺服器。
4. 該驅動程式是設定為取得 Glen 的完全可辨識 LDAP 名稱：CN=GLCanyon、OU=finance、O=Testco Financials。
5. LDAP 名稱會置於 Identity Vault 中 GCANYON 使用者的 DirXML-ADContext (使用者物件的延伸，DirXML-ADContext 的副本) 屬性。
既然 Identity Vault 中有了必要的屬性，GroupWise 驅動程式便會處理 GCANYON 物件的屬性。
6. 因為 Glen 位於「財務」部門，所以驅動程式會在「財務部門」的 GroupWise 領域伺服器上提供 GCANYON 的 GroupWise 使用者帳戶。
7. 帳戶建立成功後，GroupWise 驅動程式規則會將 Glen 的 GroupWise 驗證身分證明提供給其 eDirectory 使用者帳戶的 Secret Store。

Glen 從網際網路驗證其公司網站時，iChain 伺服器便可使用 SecretStore 填寫其驗證表單 (對象為其安全 GroupWise 電子郵件帳戶)，而無需輸入 GroupWise 身分證明，並且還對公司資源提供額外的安全性。

4.4 實作具有 SecretStore 的身分證明提供規則

可以自定具有 SecretStore 之「身分證明提供」規則實作的幅度極大。實作的步驟會視安裝 SecretStore 的平台、進行提供的應用程式和涉及的 Identity Manager 驅動程式而有所不同。

實作具有 SecretStore 的「身分證明提供」規則：

- ◆ 「符合具有 Novell SecretStore 之身分證明提供規則的要求」，第 330 頁
- ◆ 「決定 Novell SecretStore 的部署組態參數」，第 331 頁
- ◆ 「建立 Novell SecretStore 的儲存機制物件」，第 333 頁
- ◆ 「建立 Novell SecretStore 的應用程式物件」，第 340 頁
- ◆ 「設定 Novell SecretStore 的身分證明提供規則組態」，第 346 頁

4.4.1 符合具有 Novell SecretStore 之身分證明提供規則的要求

若要使用具有 SecretStore 的「身分證明提供規則」，必須備妥下列項目：

- ◆ 含支援套件 1 的 Identity Manager 3.0
 - ◆ 必須安裝於 eDirectory 8.7x；不支援 eDirectory 8.8
 - ◆ 驗證 jso.jar、idmcp.jar 和 jnet.jar 是否位於 Identity Manager Java 程式庫的標準位置
- ◆ SecretStore 3.3 或以上版本

驗證自己的環境符合要求後，請繼續進行「決定 Novell SecretStore 的部署組態參數」，第 331 頁。

4.4.2 決定 Novell SecretStore 的部署組態參數

為了提供特性 4-1 所示的部署案例中描述的同步化功能，第一步就是蒐集與 Identity Manager 和 SecretStore 環境相關的所有企業流程資訊。您可以列印表格 4-3, 「SecretStore 的身分證明提供規則工作表」, 第 331 頁，並用它做為記錄資訊的工作表。

表格 4-3 SecretStore 的身分證明提供規則工作表

需要組態資訊	資訊
1) 要設定哪些應用程式的組態，以供 Web 的「單一登入」提供使用？	
2) SecretStore 儲存機制伺服器的 DNS 名稱或 IP 位址。	
3) SecretStore 儲存機制伺服器的 SSL LDAP 連接埠。	
4) SecretStore 儲存機制伺服器之管理員的完全合法 LDAP 可辨識名稱。	
5) SecretStore 儲存機制伺服器之管理員的密碼。	
6) 從 SecretStore 伺服器輸出之 SSL 證書的完整路徑和名稱。證書必須要在 Identity Manager 伺服器本地地上。	
7) 決定是多個驅動程式會使用一個 SecretStore 儲存機制，還是每個驅動程式都會使用個別的儲存機制。	
8) 記錄正在使用的 SecretStore 機密類型。	現有兩種受支援的機密類型： <ul style="list-style-type: none">◆ A：應用程式機密 (SS_App：字首)◆ C：身分證明集機密 (SS_CredSet：字首)
9) 每個提供的應用程式之應用程式 ID 或「身分證明集」名稱。	
10) 尋找每個應用程式的所有必要驗證金鑰，例如「使用者名稱」和「密碼」。這些對每個應用程式而言可能都不同。	
11) 決定任何一個驗證金鑰值是否可與靜態值一併設定。	
12) 針對對於每個使用者而言不同或可以不同的非靜態值，記下非靜態資訊 (事件資訊或 Identity Vault 屬性值) 的來源。	
13) 如果您正在驅動程式上實作 SecretStore 提供，且驅動程式也正在將密碼同步化至目標應用程式，請決定 SecretStore 提供要在密碼於目標應用程式伺服器中設定之前或之後發生。	
14) 要在其中儲存儲存機制和應用程式物件之「驅動程式」物件的名稱。(可為不同的驅動程式。)	
15) 決定目標應用程式的「使用者」物件 DN。	

範例提供組態資料

使用者的 SecretStore 身分證明會提供給「財務」部門的 GroupWise 領域伺服器，而該身分證明所使用下列範例資料則已透過提供案例來決定，且會用於 Finance eDirectory 驗證樹中的使用者身上。

SecretStore 儲存機制資訊

表格 4-4 SecretStore 的範例身分證明提供規則工作表

需要組態資訊	資訊
1) 要設定哪些應用程式的組態，以供 Web 的「單一登入」提供使用？	GroupWise
2) SecretStore 儲存機制伺服器的 DNS 名稱或 IP 位址。	151.150.191.5
3) SecretStore 儲存機制伺服器的 SSL LDAP 連接埠。	636
4) SecretStore 儲存機制伺服器之管理員的完全合法 LDAP 可辨識名稱。	cn=admin, ou=finance, o=Tesetco Financials
5) SecretStore 儲存機制伺服器之管理員的密碼。	dixml
6) 從 SecretStore 伺服器輸出之 SSL 證書的完整路徑和名稱。證書必須要在 Identity Manager 伺服器本地上。	c:\novell\nds\FinanceAD.cer
7) 決定是多個驅動程式會使用一個 SecretStore 儲存機制，還是每個驅動程式都會使用個別的儲存機制。	此例中只有一個儲存機制。
8) 記錄正在使用的 SecretStore 機密類型。	現有兩種受支援的機密類型： <ul style="list-style-type: none">◆ A：應用程式機密 (SS_App：字首)◆ C：身分證明集機密 (SS_CredSet：字首)
9) 每個提供的應用程式的應用程式 ID 或「身分證明集」名稱。	GroupWise_Credentials
10) 尋找每個應用程式的所有必要驗證金鑰，例如「使用者名稱」和「密碼」。這些對每個應用程式而言可能都不同。	使用者名稱 密碼
11) 決定任何一個驗證金鑰值是否可與靜態值一併設定。	此案例沒有靜態資訊。
12) 針對對於每個使用者而言不同或可以不同的非靜態值，記下非靜態資訊 (事件資訊或 Identity Vault 屬性值) 的來源。	使用者名稱：Identity Vault 屬性 "CN" 密碼：事件 <password>
13) 如果您正在驅動程式上實作 SecretStore 提供，且驅動程式也正在將密碼同步化至目標應用程式，請決定 SecretStore 提供要在密碼於目標應用程式伺服器中設定之前或之後發生。	之後
14) 要在其中儲存應用程式物件和儲存機制之「驅動程式」物件的名稱。(可為不同的驅動程式。)	GroupWise-Finance 驅動程式

15) 決定目標應用程式的「使用者」物件 DN。 Identity Vault 屬性 "DirXML-ADContext"

其他環境資訊：

- ◆ 「財務」部門 eDirectory 樹會做為所有 Finance 應用程式的 SecretStore 儲存機制使用。
- ◆ 所有的財務部門提供驅動程式都位於稱為 Finance Drivers 的驅動程式集中。
- ◆ 當 Identity Vault 屬性 employeeStatus 設為值 "I" 時，必須刪除 GroupWise 帳戶，且必須從 Active Directory 使用者中移除該 GroupWise 使用者帳戶的 SecretStore 身分證明。

如蒐集的資料所示，SecretStore 儲存機制資訊廣泛適用於提供「財務」部門應用程式的所有驅動程式。此外，可以用靜態方式設定所有提供資訊的組態，但不包含 Username、Password 及 Target User DN 這些 GroupWise 登入參數。

所有的參數都決定好後，請繼續進行「[建立 Novell SecretStore 的儲存機制物件](#)」，第 333 頁。

4.4.3 建立 Novell SecretStore 的儲存機制物件

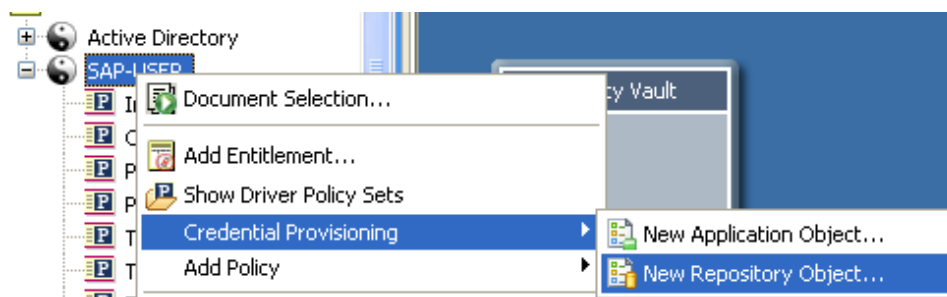
儲存機制物件會儲存 SecretStore 的靜態組態資訊。儲存機制資訊獨立於使用應用程式身分證明的應用程式以外。不論已連接系統為何（如 SAP、PeopleSoft、Notes 等），此資訊均適用於所有的提供事件。儲存機制物件可在 Designer 或 iManager 中建立。

- ◆ 「[在 Designer 中建立 Novell SecretStore 的儲存機制物件](#)」，第 333 頁
- ◆ 「[在 iManager 中建立 Novell SecretStore 的儲存機制物件](#)」，第 336 頁

在 **Designer** 中建立 **Novell SecretStore** 的儲存機制物件

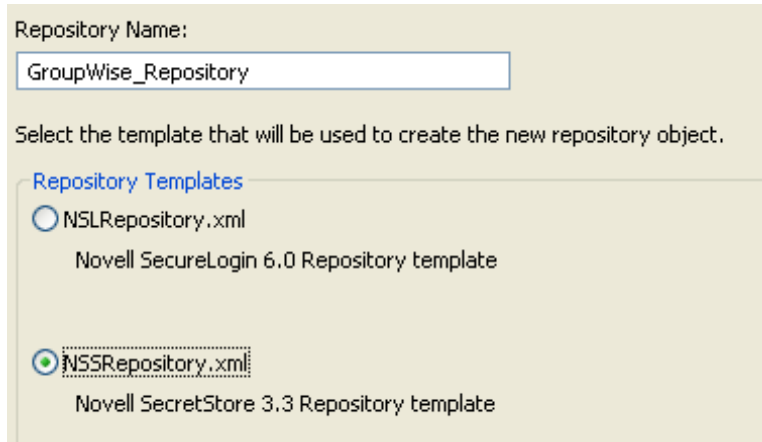
下列是您可以用來在 Designer 中建立儲存機制物件的眾多方法之一。

- 1 在大綱檢視窗中，以滑鼠右鍵按一下儲存機制物件要儲存的所在驅動程式物件。
- 2 按一下「身分證明提供 > 新儲存機制物件」。

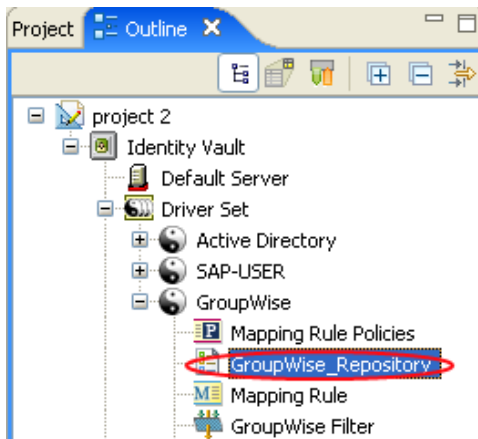


- 3 指定儲存機制物件的名稱。

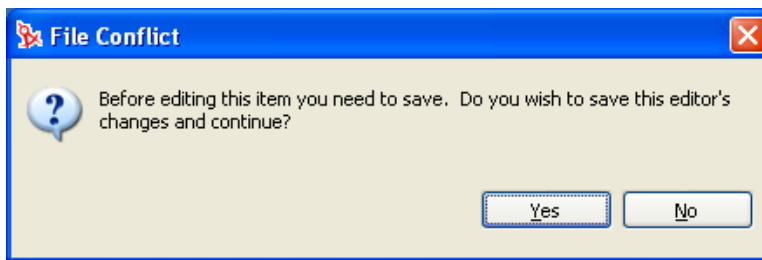
- 4 選取「NSSRepository.xml」以使用 SecretStore 範本。



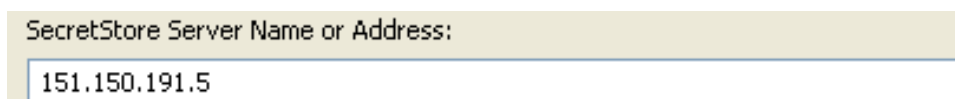
- 5 按一下「確定」。
- 6 在大綱檢視窗中連接兩下儲存機制物件，新增組態資訊。




- 7 按一下「是」儲存新儲存機制物件。



- 8 指定 SecretStore 伺服器的 DNS 名稱或 IP 位址。請參閱工作表項目 2)。



- 9 指定 SecretStore 伺服器的 SSL 連接埠。請參閱工作表項目 3)。



SecretStore Server SSL Port: 

- 10 指定從 SecretStore 伺服器輸出之 SSL 證書的完整路徑。路徑必須包含證書名稱，且必須要在 Identity Manager 伺服器本地。請參閱工作表項目 6)。

SecretStore Server SSL Certificate Path:

附註：如需如何輸出 SSL 證書的相關資訊，請參閱 iManager 文件。

- 11 指定 SecretStore 管理員的完全合法 LDAP 可辨識名稱。請參閱工作表項目 4)。

Use Enhanced Protection Flag:  


- 12 按一下「設定密碼」。


SecretStore Administrator Password:

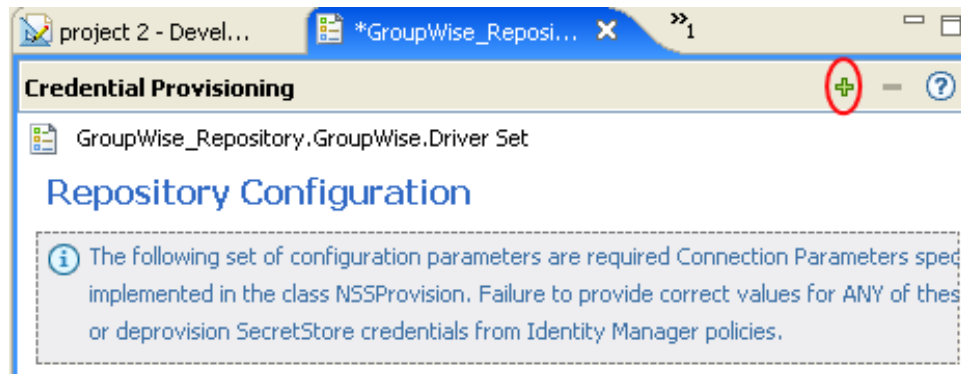
- 13 指定兩次 SecretStore 管理員的密碼，然後按一下「確定」。請參閱工作表項目 5)。



The image shows a 'Change Password' dialog box with a blue title bar and a close button. It contains two text input fields: 'Enter password:' and 'Re-enter password:'. Both fields contain six black dots representing masked characters. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

- 14 檢視資訊，然後按一下「儲存」圖示  儲存資訊。

- 15 (選擇性) 如果想要建立儲存機制物件的其他組態參數，請按一下「新增項目」圖示 。




- 15a 指定參數的名稱。
- 15b 指定參數的顯示名稱。
- 15c 指定供您參考的參數描述。
參數會儲存為字串。

Name:

Display name:

Description:

Type:
string 

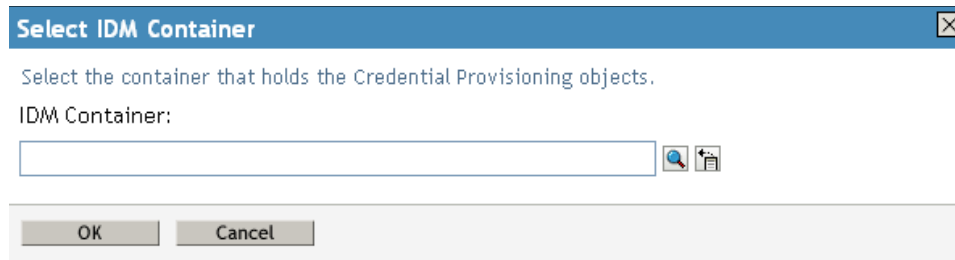
- 15d 按一下「確定」。
- 15e 按一下「儲存」圖示  儲存儲存機制物件。

儲存機制物件建立後，請繼續到「在 Designer 中建立 Novell SecureLogin 的應用程式物件」，第 319 頁。

在 iManager 中建立 **Novell SecretStore** 的儲存機制物件

- 1 在 iManager 中，選取「身分證明提供 > 組態」。

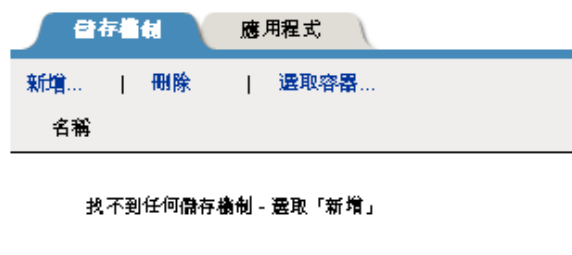
- 2 瀏覽至儲存機制物件儲存的所在「驅動程式」物件並加以選取，然後按一下「確定」。



The dialog box is titled "Select IDM Container" and contains the instruction "Select the container that holds the Credential Provisioning objects." Below this, there is a label "IDM Container:" followed by an empty text input field with search and refresh icons to its right. At the bottom, there are "OK" and "Cancel" buttons.

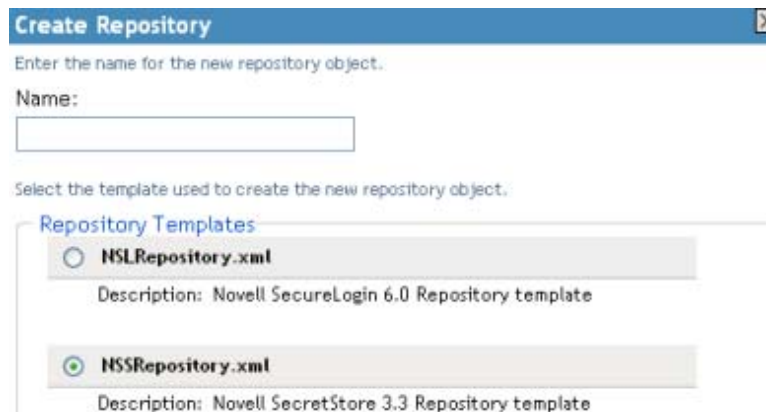
- 3 按一下新增」建立儲存機制。

IDM 容器 : GroupWise.driverset1.novell



The interface shows two tabs: "儲存機制" (Storage Mechanism) and "應用程式" (Application). Under the "儲存機制" tab, there are buttons for "新增..." (Add...), "刪除" (Delete), and "選取容器..." (Select Container...). Below these is a table with a header "名稱" (Name). The table is currently empty, and a message below it reads "找不到任何儲存機制 - 選取「新增」" (No storage mechanisms found - select "Add").


- 4 指定儲存機制物件的名稱。
- 5 選取「NSSRepository.xml」以使用 SecretStore 範本建立儲存機制。



The dialog box is titled "Create Repository" and asks to "Enter the name for the new repository object." It features a "Name:" label and an empty text input field. Below, it says "Select the template used to create the new repository object." and shows a list of "Repository Templates":

- NLSRepository.xml**
Description: Novell SecureLogin 6.0 Repository template
- NSSRepository.xml**
Description: Novell SecretStore 3.3 Repository template

- 6 按一下「確定」。
- 7 指定 SecretStore 伺服器的 DNS 名稱或 IP 位址。請參閱工作表項目 2)。

SecretStore Server Name or Address 

8 指定 SecretStore 伺服器的 SSL 連接埠。請參閱工作表項目 3)。

SecretStore Server SSL Port ⓘ

9 指定從 SecretStore 伺服器輸出之 SSL 證書的完整路徑。路徑必須包含證書名稱，且必須要在 Identity Manager 伺服器本地。請參閱工作表項目 6)。

SecretStore Server SSL Certificate Path ⓘ

附註：如需如何輸出 SSL 證書的相關資訊，請參閱 iManager 文件。

10 指定 SecretStore 管理員的完全合法 LDAP 可辨識名稱。請參閱工作表項目 4)。

SecretStore Administrator ⓘ

11 按一下「設定密碼」。

SecretStore Administrator Password ⓘ [Set password](#)

12 指定兩次 SecretStore 管理員的密碼，然後按一下「確定」。請參閱工作表項目 5)。

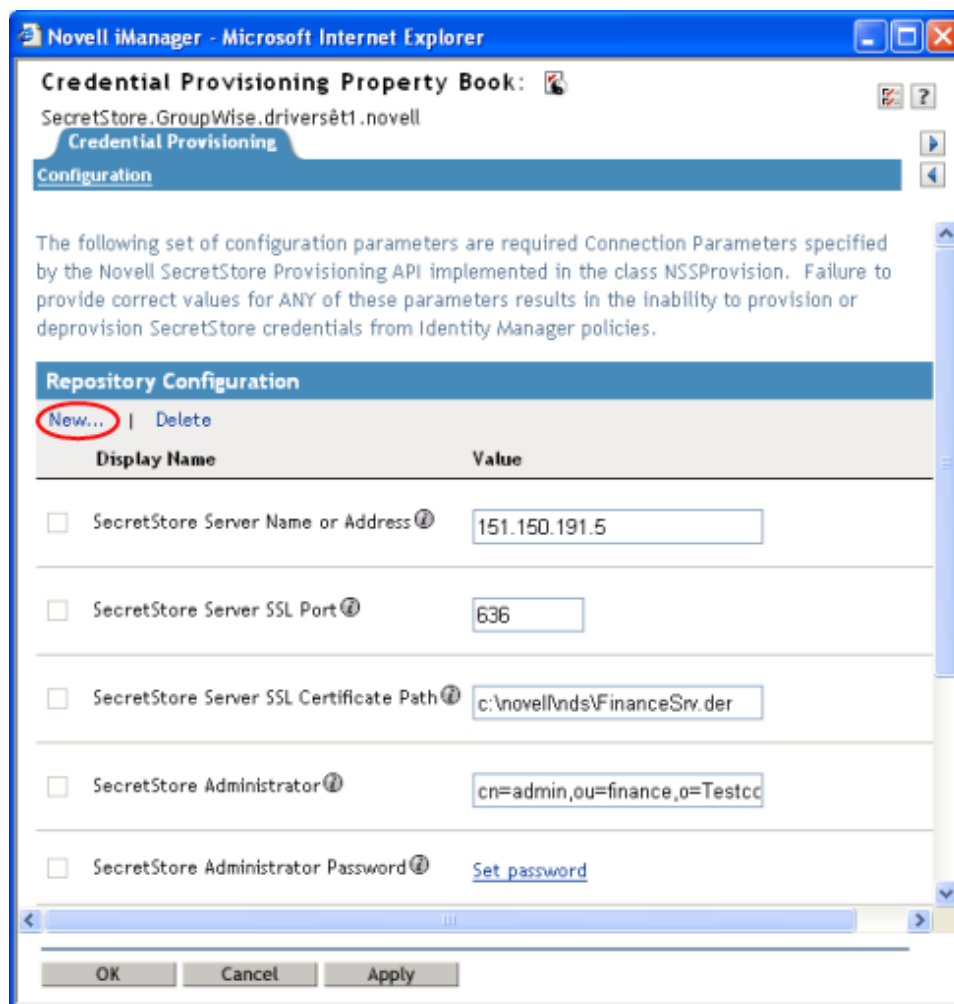
Enter Password ⓘ

Enter password:

Reenter password:

13 檢視指定的值，然後按一下「確定」。

14 (選擇性) 如果想要建立儲存機制物件的其他組態參數，請按一下「新增」。



範例資訊出自於特性 4-1 頁上 308 中的案例。

14a 指定參數的名稱。

14b 指定參數的顯示名稱。

14c 指定供您參考的參數描述。

參數會儲存為字串。

Global Configuration Value Definition

Global Configuration Values are a means through which the behavior of an Identity Manager driver configuration can be changed without requiring any policy to be changed.

Name:

Display name:

Description:

Type:
string

14d 按一下「確定」。

儲存機制物件建立後，請繼續進行「[在 iManager 中建立 Novell SecureLogin 的應用程式物件](#)」，第 322 頁。

4.4.4 建立 Novell SecretStore 的應用程式物件

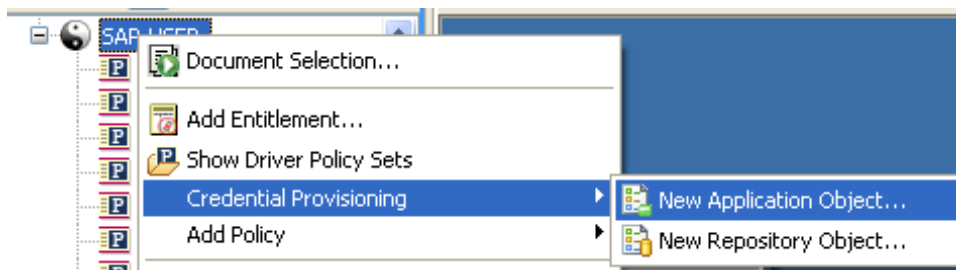
應用程式會儲存 SecretStore 的靜態組態參數。應用程式資訊是使用應用程式身分證明的應用程式特有的 (例如 GroupWise 用戶端資訊或 SAP 資料庫用戶端資訊)。應用程式物件可在 Designer 或 iManager 中建立。

- ◆ 「[在 Designer 中建立 Novell SecretStore 的應用程式物件](#)」，第 340 頁
- ◆ 「[在 iManager 中建立 Novell SecretStore 的應用程式物件](#)」，第 343 頁

在 **Designer** 中建立 **Novell SecretStore** 的應用程式物件

下列是您可以用來在 Designer 中建立應用程式物件的眾多方法之一。

- 1 在大綱檢視窗中，以滑鼠右鍵按一下應用程式物件要儲存的所在驅動程式物件。
- 2 按一下「身分證明提供 > 新應用程式物件」。

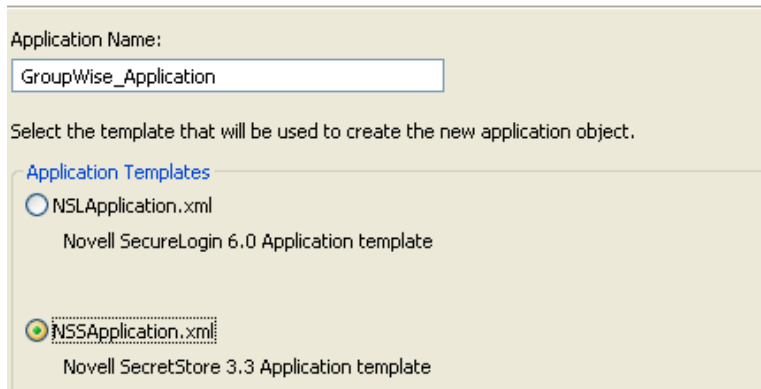


- 3 指定應用程式物件的名稱。

- 4 選取 「NSSApplication.xml」 以使用 SecretStore 範本。

Create Application

Give a name for the application object and select the default template to start with



Application Name:
GroupWise_Application

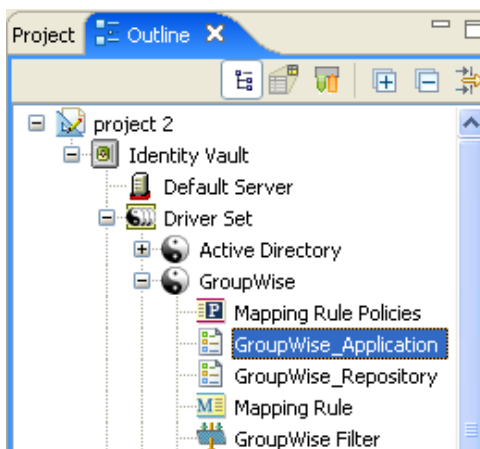
Select the template that will be used to create the new application object.

Application Templates

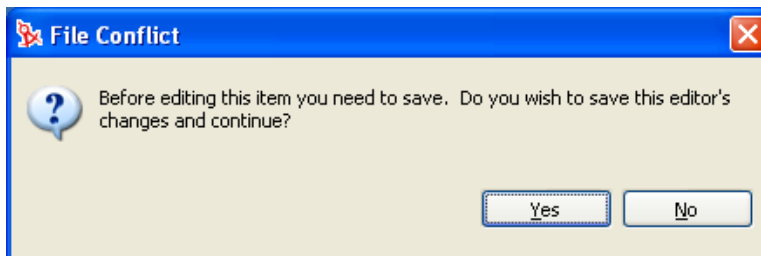
NSLApplication.xml
Novell SecureLogin 6.0 Application template

NSSApplication.xml
Novell SecretStore 3.3 Application template

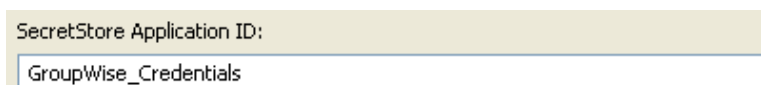
- 5 按一下「確定」。
- 6 在大綱檢視窗中連接兩下應用程式物件，新增組態資訊。



- 7 按一下「是」儲存新應用程式物件。



- 8 指定 SecretStore 應用程式 ID。請參閱工作表項目 9)。




SecretStore Application ID:
GroupWise_Credentials


- 9 選取「SecretStore 機密類型」。請參閱工作表項目 8)。

SecretStore Secret Type: Shared 


- 10 選取「SecretStore 共用機密類型」。請參閱工作表項目 8)。

SecretStore Shared Secret Type: Credential Set 

- 11 選取「SecretStore 使用增強性保護旗標」為「停用」或「啓用」。

Use Enhanced Protection Flag: Disabled 

- 12 按一下「設定密碼」設定「增強性保護密碼」(如果已啓用的話)。

Enhanced Protection Password: 

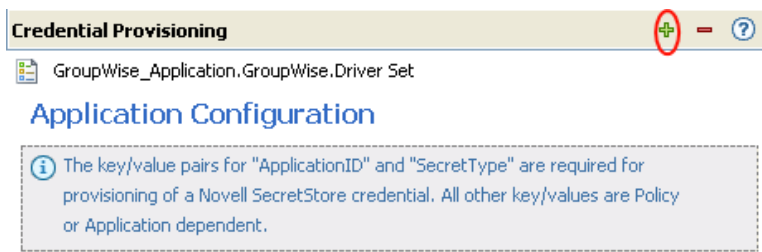
- 13 指定兩次密碼，然後按一下「確定」。



A dialog box titled "Change Password" with a close button (X) in the top right corner. It contains two text input fields. The first field is labeled "Enter password:" and contains six dots. The second field is labeled "Re-enter password:" and contains six dots. At the bottom of the dialog are two buttons: "OK" and "Cancel".

- 14 按一下「儲存」圖示  儲存應用程式。

- 15 按一下「新增項目」圖示  新增應用程式需要的驗證金鑰。



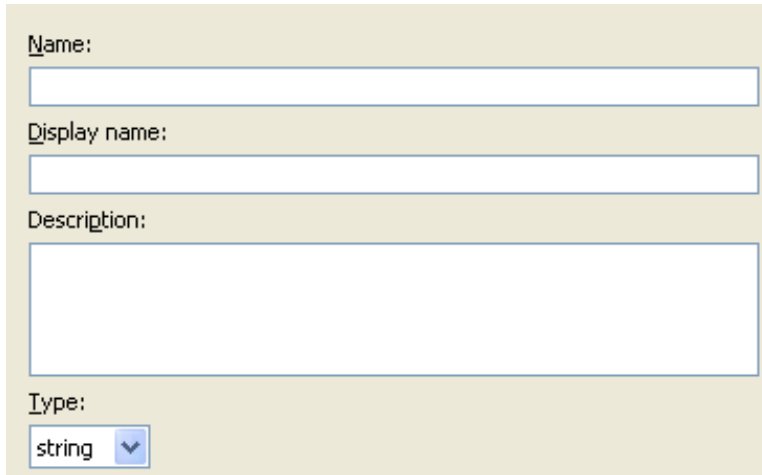
A window titled "Credential Provisioning" with a plus icon, a minus icon, and a question mark icon in the top right corner. Below the title bar, it shows "GroupWise_Application.GroupWise.Driver Set" and "Application Configuration". A dashed box contains an information icon and the text: "The key/value pairs for 'ApplicationID' and 'SecretType' are required for provisioning of a Novell SecretStore credential. All other key/values are Policy or Application dependent."

- 15a 指定驗證金鑰的名稱。

- 15b 指定驗證金鑰的顯示名稱。

- 15c 指定供您參考的驗證金鑰描述。

驗證金鑰會儲存為字串。



Name:

Display name:

Description:

Type:
string ▼

15d 按一下「確定」。

15e 針對每個需要輸入的新驗證金鑰，重複**步驟 15**。

16 如果驗證金鑰是所有使用者身分證明所共用的靜態值，則指定驗證金鑰。

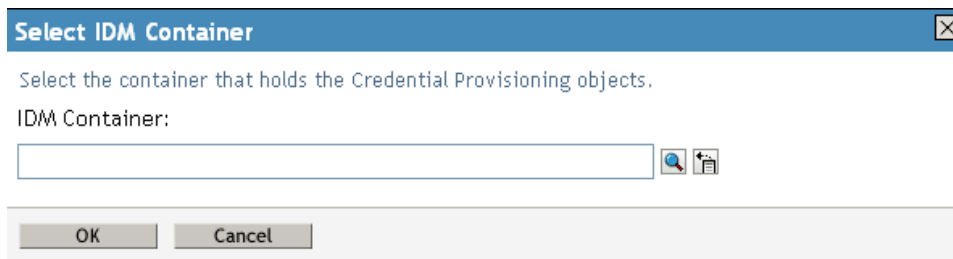
17 按一下「儲存」圖示  儲存應用程式。

應用程式物件建立後，請繼續進行「設定 Novell SecretStore 的身分證明提供規則組態」，[第 346 頁](#)。

在 **iManager** 中建立 **Novell SecretStore** 的應用程式物件

1 在 iManager 中，選取「身分證明提供 > 組態」。

2 瀏覽至應用程式物件儲存的所在「驅動程式」物件並加以選取，然後按一下「確定」。



Select IDM Container

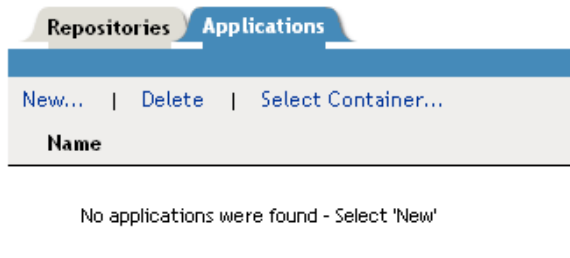
Select the container that holds the Credential Provisioning objects.

IDM Container:

OK Cancel

3 選取「應用程式」索引標籤，然後按一下「新增」。

Container: Delimited Text.DriverSet.Novell



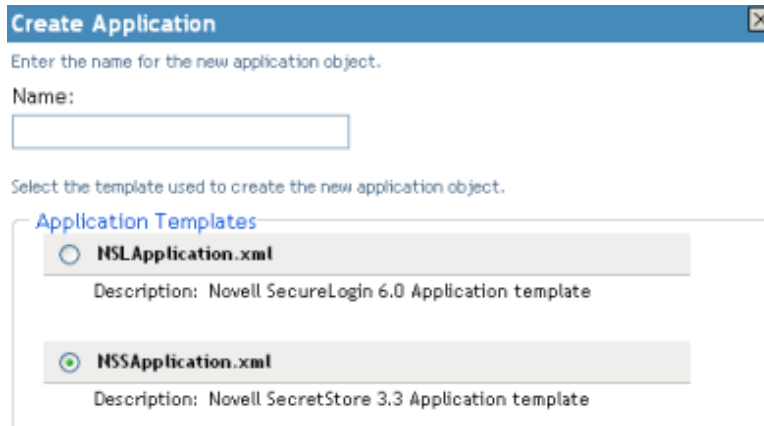
Repositories Applications

New... | Delete | Select Container...

Name

No applications were found - Select 'New'

- 指定應用程式物件的名稱
- 選取「*NSSApplication.xml*」以使用 SecretStore 範本建立應用程式。



- 按一下「確定」。
- 指定「*SecretStore* 應用程式 ID」。請參閱工作表項目 9)。

SecretStore Application ID ⓘ

- 選取「*SecretStore* 機密類型」。請參閱工作表項目 7)。*SecretStore* 類型為「共用」或「不共用」。

SecretStore Secret Type ⓘ Shared ▼

- 選取「*SecretStore* 共用機密類型」。請參閱工作表項目 8)。「共用」的 *SecretStore* 類型為「身分證明集」或「應用程式」。

SecretStore Shared Secret Type ⓘ Credential Set ▼

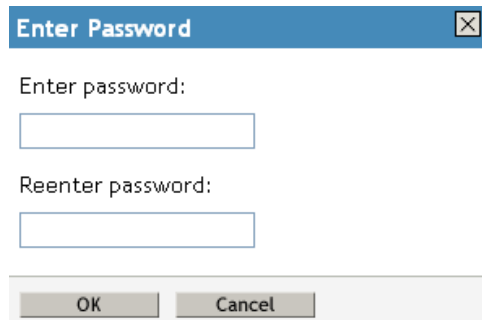
- 選取 *SecretStore* 「使用增強性保護旗標」為「停用」或「啓用」。

Use Enhanced Protection Flag ⓘ Disabled ▼

- 按一下「設定密碼」設定「增強性保護密碼」(如果已啓用的話)。

Enhanced Protection Password ⓘ [Set password](#)

12 指定兩次密碼，然後按一下「確定」。



Enter Password

Enter password:

Reenter password:

OK Cancel

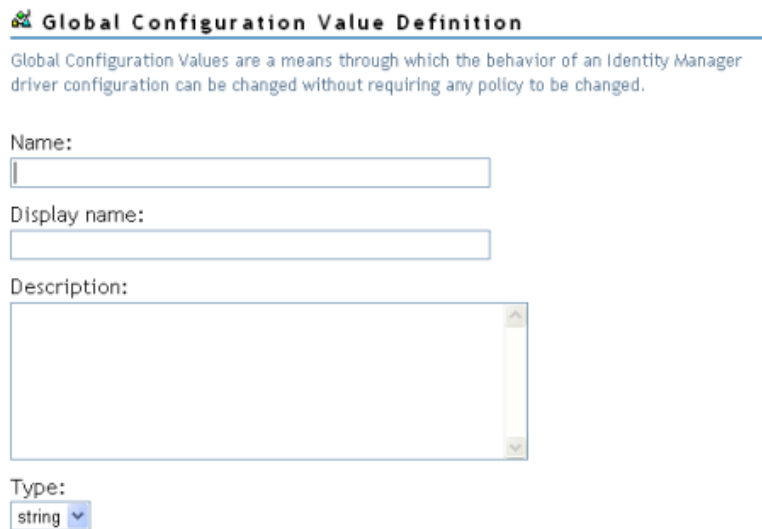
13 按一下「新增」建立應用程式需要的驗證金鑰。請參閱工作表項目 10)。

13a 指定驗證金鑰的名稱。

13b 指定驗證金鑰的顯示名稱。

13c 指定供您參考的驗證金鑰描述。

驗證金鑰會儲存為字串。



Global Configuration Value Definition

Global Configuration Values are a means through which the behavior of an Identity Manager driver configuration can be changed without requiring any policy to be changed.

Name:

Display name:

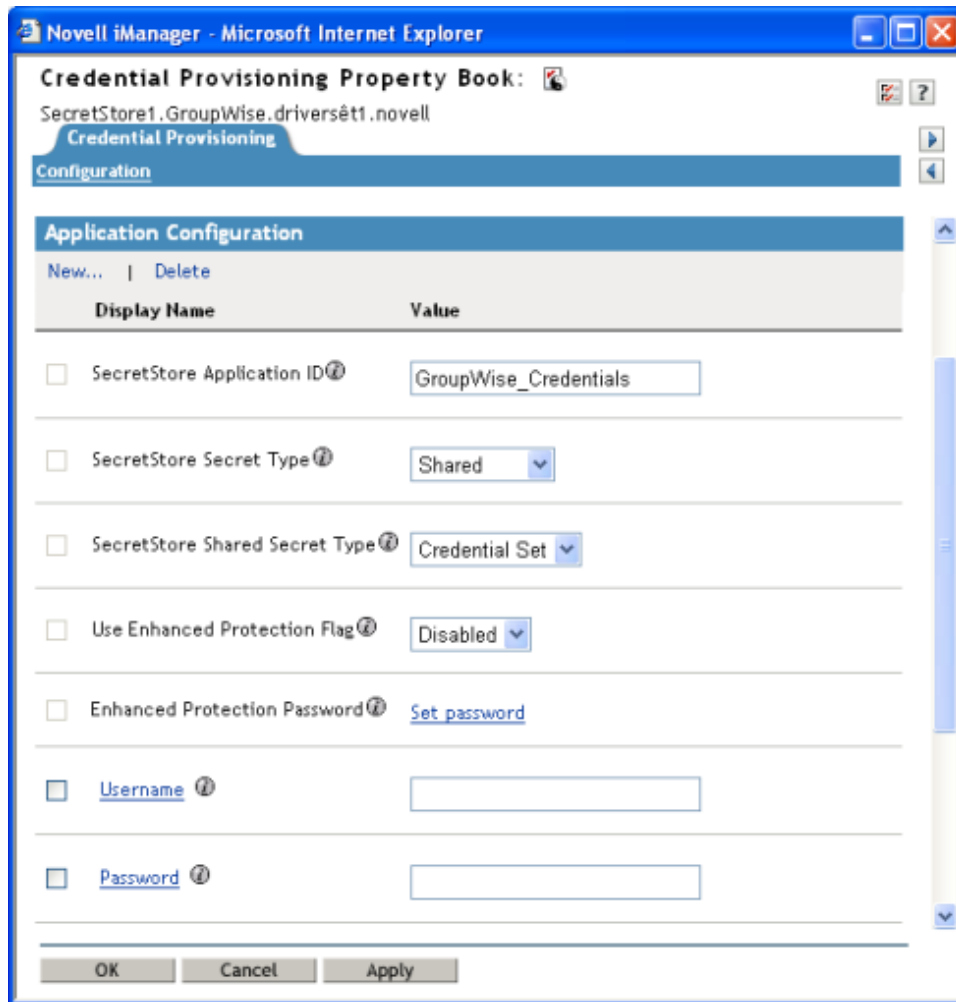
Description:

Type:
string

13d 按一下「確定」。

13e 如需應用程式需要的驗證金鑰，請重複步驟 13。

- 14 如果驗證金鑰值為靜態值，則指定該值，然後按一下「確定」。



應用程式物件建立後，請繼續進行「設定 Novell SecretStore 的身分證明提供規則組態」，第 346 頁。

4.4.5 設定 Novell SecretStore 的身分證明提供規則組態

儲存機制物件和應用程式物件建立後，便需要建立規則以提供 SecretStore 資訊。規則會使用儲存於儲存機制和應用程式物件中的資訊。「規則產生器」中有兩種動作，可以提供 SecretStore 身分證明：

- ◆ 「清除 SSO 身分證明」，第 325 頁
- ◆ 「設定 SSO 身分證明」，第 326 頁

清除 SSO 身分證明

「清除 SSO 身分證明」動作可讓您清除 SSO 身分證明，所以便可取消提供物件。

特性 4-6 清除 SSO 身分證明

The screenshot shows the 'Action List' dialog box with the action 'clear SSO credential' selected. The dialog contains several input fields and a checkbox:

- Do:** clear SSO credential
- Enter credential store object DN:*** (with a search icon)
- Render browsed DN relative to policy**
- Enter target user DN:*** (with a search icon)
- [Populate the following from an application object](#)
- Enter application credential ID:***
- Enter login parameter strings:** (with a search icon)

- ◆ 輸入身分證明儲存物件 **DN**：瀏覽至儲存機制物件並加以選取。
- ◆ 輸入目標使用者 **DN**：使用「引數產生器」建立目標使用者 DN。請參閱工作表項目 15)。
- ◆ 輸入應用程式身分證明 **ID**：指定應用程式 ID。請參閱工作表項目 9)。
- ◆ 輸入登入參數字串：啟動「字串產生器」並輸入應用程式的每個驗證金鑰。請參閱工作表項目 10)。

設定 SSO 身分證明

「設定 SSO 身分證明」動作可讓您在使用者物件建立或密碼經過修改時，設定 SSO 身分證明。

特性 4-7 設定 SSO 身分證明

The screenshot shows the 'Action List' dialog box with the action 'set SSO credential' selected. The dialog contains several input fields and a checkbox:

- Do:** set SSO credential
- Enter credential store object DN:*** (with a search icon)
- Render browsed DN relative to policy**
- Enter target user DN:*** (with a search icon)
- [Populate the following from an application object](#)
- Enter application credential ID:***
- Enter login parameter strings:** (with a search icon)

- ◆ 輸入身分證明儲存物件 **DN**：瀏覽至儲存機制物件並加以選取。
- ◆ 輸入目標使用者 **DN**：使用「引數產生器」建立目標使用者 DN。請參閱工作表項目 15)。
- ◆ 輸入應用程式身分證明 **ID**：指定應用程式 ID。請參閱工作表項目 9)。
- ◆ 輸入登入參數字串：啟動「字串產生器」並輸入應用程式的每個驗證金鑰。請參閱工作表項目 10)。

範例身分證明提供規則

您可以實作並自定身分證明提供規則，以符合環境需求。下列範例說明如何實作[特性 4-5 頁上 329](#) 中呈現之案例的規則。

在「財務」案例中，SecretStore 提供會發生在 GroupWise 中的密碼成功設定以後。大多數的必要參數組態都是以靜態方式設定，而且都可以透過存機制物件和應用程式物件，供所有的規則使用。不過，有些資料參數並非靜態 (CN、password 及 DirXML-ADContext)，而且只有在 GroupWise 使用者 <add> 或 <modify-password> 指令完成，且 GroupWise 驅動程式 Shim 傳回 <output> 狀態文件後才可使用。<output> 文件不再包含任何「訂閱者」操作屬性，且指令的「使用者」網路位置也遺失，因此阻止了對於物件的查詢。因此有必要進行下列幾項操作：

- ◆ 確定 GroupWise 驅動程式的「訂閱者建立」規則強制非靜態資料參數存在。
- ◆ 將「訂閱者」指令發出至 GroupWise 驅動程式 Shim 之前，先快取提供操作所需的非靜態參數。
- ◆ 指令成功完成後取回已快取的資料，以便用於 SecretStore 提供。

附註：Identity Manager 3.0 支援套件 1 的媒體上有 XML 格式的範例規則。檔名為 SampleInputTransform.xml、SampleSubCommandTransform.xml 及 SampleSubEventTransform.xml。這些檔案位於下列目錄中：

- ◆ linux\setup\utilities\cred_prov
- ◆ nt\dirxml\utilities\cred_prov
- ◆ nw\dirxml\utilities\cred_prov

如果在安裝公用程式時選取「身分證明提供範例規則」，這些檔案便會安裝到 Identity Manager 伺服器。範例規則會安裝到下列位置 (視平台而定)：

- ◆ Windows：C:\Novell\NDS\DirXMLUtilities (此為預設值；使用者於安裝時可加以變更)
- ◆ NetWare：SYS:\System\DirXmlUtilities
- ◆ Linux (eDir 8.7)：/usr/lib/dirxml/rules/credprov

範例規則提供您一個起點，讓您開發用於個人環境的規則。

操作資料快取

可供所需操作資料之快取使用的機制為 <operation-data> 元素。因為您可能需要從 <add> 或 <modify-password> 指令提供 SecretStore 帳戶，所以實作非靜態資料快取規則的合理位置就是在「訂閱者指令轉換」規則中。下列範例顯示典型的「SecretStore 提供」元素：

```
<operation-data> <nss-sync-data> <nss-target-user-dn> cn=GLCANYON,ou=finance,o=Testco Financials </nss-target-user-dn> <nss-app-username>GCANYON</nsl-app-username> <password><!-- content suppressed --></password> <nss-passphrase-answer>50024222</nsl-passphrase-answer> </nss-sync-data> </operation-data>
```

在來自於[特性 4-5 頁上 329](#) 的範例「財務」部門案例中，若要填入操作資料封包內容，則需要下列這些值：

- ◆ <nss-target-user-dn> 元素會從 Identity Vault 中，以 DirXML-ADContext 屬性值加以填入，該值則已由 eDirectory 驅動程式設定。若要確保 eDirectory 驅動程式設定該值時，

GroupWise 驅動程式會收到通知，請務必將 DirXML-ADContext 做為通知屬性新增至「訂閱者」過濾器。

- ◆ <nss-app-username> 元素會由 Identity Vault 中的 CN 屬性值填入。
- ◆ password 元素會由 <add> 或 <modify-password> 指令中的 <password> 元素值填入。

SecretStore 提供

在此範例案例中，可將操作資料從其中取回，並用於 SecretStore 身分證明提供的第一個可用位置就是在驅動程式的「輸出轉換」規則中。範例案例中實作了兩種規則：

- ◆ 密碼同步化成功後，設定「SecretStore 身分證明」
- ◆ 如果應用程式已刪除 (Identity Vault 物件未刪除)，則移除「SecretStore 身分證明」

附註：SampleInputTransform.xml 檔案中有範例規則，可在密碼同步化成功後設定 SecretStore 身分證明。該檔案位於 Identity Manager 3.0 支援套件 1 的媒體上，公用程式目錄的 cred_prov 資料夾中。

「設定 SecretStore 身分證明」規則需要確定只有在傳回的指令狀態為 Success，且之前設定的 <operation-data> 存在時，提供才會發生。

SecretStore 取消提供

有許多案例可以使用規則，其中已連接應用程式的使用者帳戶已刪除，而 Identity Vault 帳戶則予以保留。在「財務」案例中，當使用者的 Identity Vault 屬性值 semployeeStatus 設為 "I" 時，則有必要刪除 GroupWise 帳戶並取消提供 SecretStore 身分證明。為了處理此情況，GroupWise 驅動程式的「訂閱者事件轉換」包含要將修改屬性值轉換為物件刪除的規則。因為刪除指令完成後仍需要 eDirectory 帳戶名稱，所以 <operation-data> 事件需要在 <delete> 指令進行時設定，如此便可在「輸入轉換」規則中供 SecretStore 取消提供規則使用。

```
<operation-data> <nss-sync-data> <nss-target-user-dn>cn=GLCANYON,ou=finance,o=Testco  
Financials </nss-targer-user-dn> </nss-sync-data> </operation-data>
```

在 Identity Manager 3.0 支援套件 1 的媒體上，位於 utilities 目錄的 SampleSubEventTransform.xml 檔案中用於將 <modify> 事件轉換為 <delete> 並建立此元素的規則 (其格式為 XML)。

使用 XSLT 樣式表定義規則

規則可以實作為 XSLT 樣式表。XSLT 是用於轉換 XML 文件的一種標準語言。Metadirectory 引擎中的 XSLT 處理器與 W3C 建議 (1999 年 11 月 16 日) 相容。如需相關規格，請參閱下列項目：

- ◆ XSL 轉換 (XSLT) (<http://www.w3.org/TR/1999/REC-xslt-19991116>)
- ◆ XML 路徑語言 (XPath) (<http://www.w3.org/TR/1999/REC-xpath-19991116>)

下列各節描述搭配 Identity Manager 使用樣式表的實作詳細資料。

- ◆ 「在 Designer 中管理 XSLT 樣式表」，第 351 頁
- ◆ 「在 iManager 中管理 XSLT 樣式表」，第 353 頁
- ◆ 「開始使用身分轉換」，第 354 頁
- ◆ 「使用 Identity Manager 傳遞的參數」，第 354 頁
- ◆ 「使用延伸功能」，第 356 頁
- ◆ 「建立密碼範例：建立規則」，第 357 頁
- ◆ 「建立 eDirectory 使用者範例：建立規則」，第 358 頁

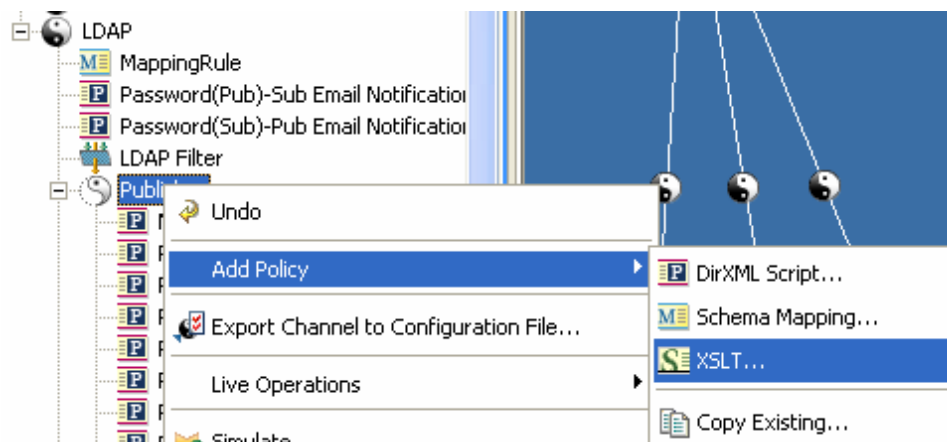
5.1 在 Designer 中管理 XSLT 樣式表

您可使用 Designer 新增、修改及刪除 XSLT 規則樣式表。下列各節提供在 Designer 中使用 XSLT 樣式表的詳細資料：

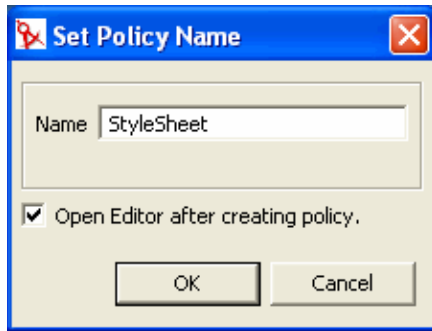
- ◆ 「在 Designer 中新增 XSLT 規則」，第 351 頁

5.1.1 在 Designer 中新增 XSLT 規則

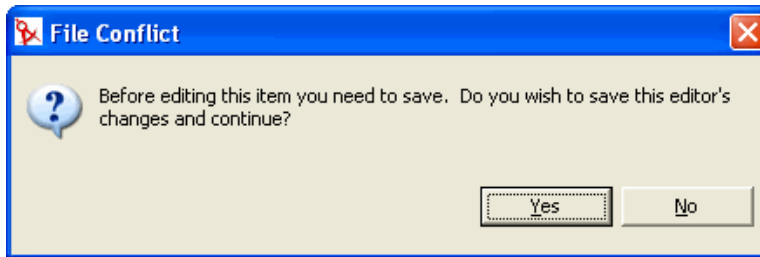
- 1 在 Designer 中開啓專案，並選取「大綱」索引標籤。
- 2 選取要在其中擁有樣式表的驅動程式和位置。
- 3 按一下滑鼠右鍵，並選取「新增規則 > XSLT」。



- 4 指定樣式表的名稱。
- 5 選取「建立規則後開啓編輯器」，然後按一下「確定」。



- 6 編輯新規則前，選取「是」儲存專案。



- 7 在 add your custom templates here 這一行字底下新增樣式表資訊。



- 8 選取「檔案 > 儲存」以儲存樣式表。


5.2 在 iManager 中管理 XSLT 樣式表

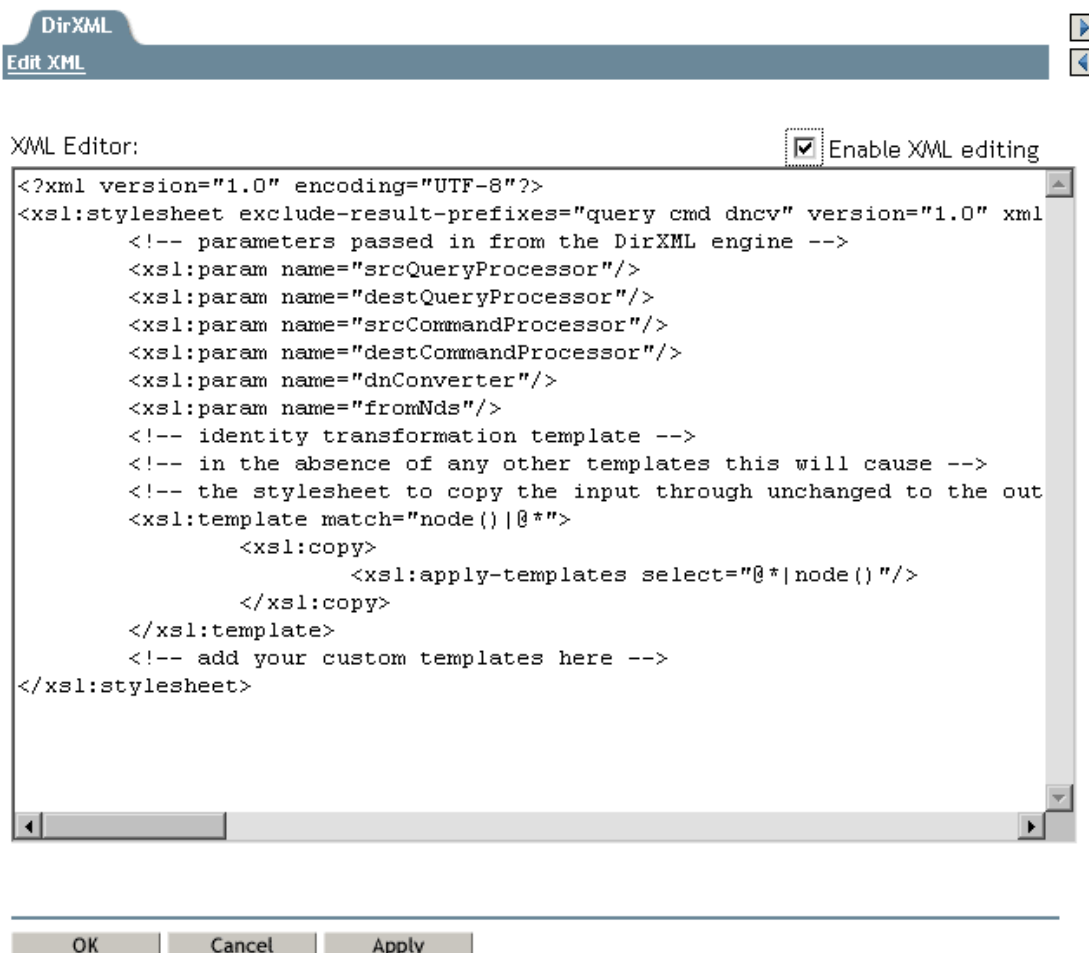
您可使用 iManager 新增、修改及刪除 XSLT 規則樣式表。下列各節提供在 iManager 中使用 XSLT 樣式表的詳細資料：

- 「在 iManager 中新增 XSLT 規則」，第 353 頁

5.2.1 在 iManager 中新增 XSLT 規則

- 1 開啓想要管理之驅動程式的「Identity Manager 驅動程式概觀」。
- 2 按一下代表要定義之規則的圖示。
- 3 按一下「插入」。
- 4 爲新規則提供名稱，然後選取 XSLT，再按一下「輸入」。
- 5 定義 XSLT 規則，然後按一下「確定」：

DirXML Policy:  xslt policy



5.3 開始使用身分轉換

在 iManager 或 Designer 中建立新樣式表時，會以實作身分轉換的樣式表預先填入。如果沒有其他範本，身分轉換便會在樣式表不變更的狀態下，透過樣式表輸入 XML 文件。實作的規則的方式通常是藉由新增其他範本，以便只處理要變更的 XML。如果您正要使用樣式表將文件從有別於 XDS (如 SOAP 和「分隔文字」驅動程式的「輸入和輸出轉換」) 的 XML 詞彙轉換為文件，或是將文件轉換為該詞彙，則您便需要移除身分範本。

5.4 使用 Identity Manager 傳遞的參數

Metadirectory 引擎會將樣式表可使用的下列參數傳遞給規則樣式表。

- ◆ `srcQueryProcessor`--- 實作 `XdsQueryProcessor` 介面的一種 Java 物件。此物件可讓樣式表查詢來源資料庫，以取得相關資訊。
- ◆ `destQueryProcessor`--- 實作 `XdsQueryProcessor` 介面的一種 Java 物件。此物件可讓樣式表查詢目的資料儲存，以取得相關資訊。
- ◆ `srcCommandProcessor`--- 實作 `XdsCommandProcessor` 介面的一種 Java 物件。此物件可讓樣式表將指令寫回事件來源。DirXML 1.0 中沒有。
- ◆ `destCommandProcessor`--- 實作 `XdsCommandProcessor` 介面的一種 Java 物件。此物件可讓樣式表直接發出指令，以便將指令傳送至目的資料儲存。
- ◆ `dnConverter`--- 此為實作 `XdsCommandProcessor` 介面的一種 Java 物件。此物件可讓樣式表將 Identity Vault 物件 DN 從一種格式轉換為另一種格式。如需相關資訊，請參閱 [介面 DNConverter \(http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/api/com/novell/nds/dirxml/driver/DNConverter.html\)](http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/api/com/novell/nds/dirxml/driver/DNConverter.html)。
- ◆ `fromNds`--- 此為一種布林值，且如果來源資料儲存為 Identity Vault，則該值為 `true`；如果為已連接應用程式，則為 `false`。

在 iManager 或 Designer 中建立新樣式表時，會以包含這些參數之宣告的樣式表預先填入。

搭配綱要映射規則、輸入轉換規則和輸出轉換規則使用查詢和指令參數時，會套用下列限制：

- ◆ 發出到應用程式 Shim 的查詢，其格式必須為該應用程式 Shim 所需的格式。換句話說，綱要名稱必須位於應用程式名稱空間，且查詢必須符合 Shim 原始使用的任何 XML 詞彙。沒有關聯參考會新增至查詢。
- ◆ 來自應用程式 Shim 的回應是使用未執行修改或綱要映射，且未解析關聯參考時，該 Shim 所傳回的格式。
- ◆ 發出到 eDirectory™ 查詢，其格式必須為 eDirectory 所需的格式。換句話說，綱要名稱必須位於 eDirectory 名稱空間，而且查詢必須為 XDS。關聯參考不會經過解析。
- ◆ 來自應用程式 Shim 的回應是使用未執行修改或綱要映射時，該 Shim 所傳回的格式。

查詢處理器

查詢處理器的使用會視 Novell® XSLT 的延伸功能實作而定。若要進行查詢，則需宣告 `XdsQueryProcessor` 介面的名稱空間。做法則是將下列項目新增至樣式表的 `<xsl:stylesheet>` 或 `<xsl:transform>` 元素。

```
xmlns:query="http://www.novell.com/nxsl/java/
```

```
com.novell.nds.dirxml.driver.XdsQueryProcessor"
```

在 iManager 或 Designer 中建立新樣式表時，會以名稱空間宣告預先填入。如需查詢處理器的相關資訊，請參閱類別 `XdsQueryProcessor` (<http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/api/com/novell/nds/dirxml/driver/XdsQueryProcessor.html>)

下列範例使用其中一種查詢處理器 (特別長的文字行已換行，且不以 < 做為開頭)：

```
<!-- Query object name queries NDS for the passed object name -->

<xsl:template name="query-object-name">
  <xsl:param name="object-name"/>

  <!-- build an xds query as a result tree fragment -->
  <xsl:variable name="query">
    <query>
      <search-class class-name="{ancestor-or-self:
        :add/@class-name}"/>

      <!-- NOTE: depends on CN being the naming attribute -->
      <search-attr attr-name="CN">
        <value><xsl:value-of select="$object-name"/
          ></value>
      </search-attr>

      <!-- put an empty read attribute in so that we don't get -->
      <!-- the whole object back -->
      <read-attr/>
    </query>
  </xsl:variable>

  <!-- query NDS -->
  <xsl:variable name="result" select="query:query($destQuery
    Processor,$query)"/>

  <!-- return an empty or non-empty result tree fragment -->
  <!-- depending on result of query -->
  <xsl:value-of select="$result//instance"/>
</xsl:template>
```

這裡是另一個範例。

```
<?xml version="1.0"?>
<xsl:transform
  version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:cmd="http://www.novell.com/nxsl/java
  com.novell.nds.dirxml.driver.XdsCommandProcessor"
>
<xsl:param name="srcCommandProcessor"/>

<xsl:template match="node()|@*">
  <xsl:copy>
```

```

        <xsl:apply-templates select="@*|node()"/>
    </xsl:copy>
</xsl:template>

<xsl:template match="add">
    <xsl:copy>
        <xsl:apply-templates select="@*|node()"/>
    </xsl:copy>

    <!-- on a user add, add Engineering department to the source
object -->
    <xsl:variable name="dummy">
        <modify class-name="{@class-name} "dest-dn="{@src-dn}">
            <xsl-copy-of select="association"/>
            <modify-attr attr-name="OU">
                <add-value>
                    <value type="string">Engineering</value>
                </add-value>
            </modify-attr>
        </modify>
    </xsl:variable>
    <xsl:variable name="dummy2"
        select="cmd:execute($srcCommandProcessor, $dummy)"/>
</xsl:template>

</xsl:transform>

```

5.5 使用延伸功能

如果是用於執行某些種類的轉換，XSLT 是一套極佳的工具，但若是用於執行其他種類的轉換（如非一般的字串處理和重覆性程序），則是相當差的工具。所幸 Novell XSLT 處理器會實作延伸功能，讓樣式表可以呼叫在 Java 中實作的功能，以及呼叫（藉由延伸的方式）任何可透過 JNI 存取的其他語言。

如需具體範例，請參閱使用查詢處理器的上述範例，以及使用 Java 處理字串的下列圖例（特別長的文字行已換行，且不以 < 做為開頭）。

```

<!-- get-dn-prefix places the part of the passed dn that -->
<!-- precedes the last occurrence of '\ ' in the passed dn -->
<!-- in a result tree fragment meaning that it can be -->
<!-- used to assign a variable value -->

<xsl:template name="get-dn-prefix" xmlns:jstring="http://
    www.novell.com/nxsl/java/java.lang.String">

    <xsl:param name="src-dn"/>

    <!-- use java string stuff to make this much easier -->
    <xsl:variable name="dn" select="jstring:new($src-dn)"/>
    <xsl:variable name="index" select="jstring:lastIndexOf
        ($dn, '\ ')" />
    <xsl:if test="$index != -1">

```

```

        <xsl:value-of select="jstring:substring($dn,0,$index)
        "/>
    </xsl:if>
</xsl:template>

```

5.6 建立密碼範例：建立規則

下列樣式表可用於「建立」規則。此規則會建立使用者、從使用者的 Surname 和 CN 屬性產生使用者的密碼，以及執行身分轉換 (透過文件中除了您嘗試解譯和轉換以外的任何項目傳遞)。

```

<?xml version="1.0" encoding="ISO-8859-1"?>

<!-- This stylesheet has an example of how to replace a create rule
with
      an XSLT stylesheet and supply an initial password for "User"
objects. -->

<xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform
      "version="1.0">

<!-- ensure we have required NDS attributes -->
<xsl:template match="add">
    <xsl:if test="add-attr[@attr-name='Surname'] and
        add-attr[@attr-name='CN']">
        <!-- copy the add through -->
        <xsl:copy>
            <xsl:apply-templates select="@*|node()"/>
            <!-- add a <password> element -->
            <xsl:call-template name="create-password"/>
        </xsl:copy>
    </xsl:if>

<!-- if the xsl:if fails, we don't have all the required attributes
so we won't copy the add through, and the create rule will veto
the add -->

</xsl:template>

<xsl:template name="create-password">
    <password>
        <xsl:value-of select="concat(add-attr[@attr-name='Surname']/
value,
        '- ',add-attr[@attr-name='CN']/value)"/>
    </password>
</xsl:template>

<!-- identity transform for everything we don't want to change -->

<xsl:template match="@*|node() ">
    <xsl:copy>
        <xsl:apply-templates select="@*|node()"/>

```



```

    </xsl:copy>
</xsl:template>

</xsl:transform>

```

5.7 建立 eDirectory 使用者範例：建立規則

此樣式表可用於「建立」規則，並顯示如何從外部應用程式中建立的項目來建立 eDirectory 使用者。此範例的概念基礎是先在「人力資源」資料庫中建立新僱用的人員，再於網路上加以建立。這需要使用者的名和姓，且會在 eDirectory 網路樹中產生唯一的 CN。雖然 eDirectory 要求 CN 只在容器中是唯一的，此樣式表仍會確保其在 eDirectory 網路樹的所有容器中都是唯一的。

```

<?xml version="1.0" encoding="ISO-8859-1"?>

<!-- This stylesheet is an example of how to replace a create rule
with an
      XSLT stylesheet and that creates the User name from the Surname
and
      given Name attributes -->

<xsl:transform
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"
  xmlns:query="http://www.novell.com/nxsl/java/
com.novell.nds.dirxml.driver.
      XdsQueryProcessor"
  >

<!-- This is for testing the stylesheet outside of Identity Manager so
things
      are pretty to look at -->
<xsl:strip-space elements="*" />
<xsl:preserve-space elements="value,component" />
<xsl:output method="xml" indent="yes" />

<!-- Identity Manager always passes two stylesheet parameters to an
XSLT rule:
      an inbound and outbound query processor -->
<xsl:param name="srcQueryProcessor" />
<xsl:param name="destQueryProcessor" />

<!-- match <add> elements -->
<xsl:template match="add">

  <!-- ensure we have required NDS attributes we need for the name -->
  <xsl:if test="add-attr[@attr-name='Surname'] and
      add-attr[@attr-name='Given Name']">

    <!-- copy the add through -->
    <xsl:copy>
      <!-- copy any attributes through except for the src-dn -->
      <!-- we'll construct the src-dn below so that the placement

```

```

rule will work -->
    <xsl:apply-templates select="@*[string(.) != 'src-dn']"/>

    <!-- call a template to construct the object name and place the
result in a variable -->
    <xsl:variable name="object-name">
        <xsl:call-template name="create-object-name"/>
    </xsl:variable>

    <!-- now create the src-dn attribute with the created name -->
    <xsl:attribute name="src-dn">
        <xsl:variable name="prefix">
            <xsl:call-template name="get-dn-prefix">
                <xsl:with-param name="src-dn" select="string(@src-
dn)"/>
            </xsl:call-template>
        </xsl:variable>
        <xsl:value-of select="concat($prefix,'\',$object-name)"/>
    </xsl:attribute>

    <!-- if we have a "CN" attribute, set it to the constructed
name -->
    <xsl:if test="./add-attr[@attr-name='CN']">
        <add-attr attr-name="CN">
            <value type="string"><xsl:value-of select="$object-
name"/></value>
        </add-attr>
    </xsl:if>

    <!-- copy the rest of the stuff through, except for what we
have already copied -->
    <xsl:apply-templates select="*[name() != 'add-attr' or @attr-
name != 'CN'] |
                                comment() |
                                processing-instruction() |
                                text()"/>

    <!-- add a <password> element -->
    <xsl:call-template name="create-password"/>

    </xsl:copy>
</xsl:if>
    <!-- if the xsl:if fails, it means we don't have all the required
attributes
so we won't copy the add through, and the create rule will veto
the add -->
</xsl:template>

<!-- get-dn-prefix places the part of the passed dn that precedes the
-->
<!-- last occurrence of '\ ' in the passed dn in a result tree fragment
-->
<!-- meaning that it can be used to assign a variable value
-->

```

```

<xsl:template name="get-dn-prefix" xmlns:jstring="http://
www.novell.com/nxsl/java/java.lang.String">
  <xsl:param name="src-dn"/>

  <!-- use java string stuff to make this much easier -->
  <xsl:variable name="dn" select="jstring:new($src-dn)"/>
  <xsl:variable name="index" select="jstring:indexOf($dn,'\')"/>
  <xsl:if test="$index != -1">
    <xsl:value-of select="jstring:substring($dn,0,$index)"/>
  </xsl:if>
</xsl:template>

<!-- create-object-name creates a name for the user object and places
the -->
<!-- result in a result tree fragment
-->
<xsl:template name="create-object-name">

  <!-- first try is first initial followed by surname -->
  <xsl:variable name="given-name" select="add-attr[@attr-name='Given
Name']/value"/>
  <xsl:variable name="surname" select="add-attr[@attr-
name='Surname']/value"/>
  <xsl:variable name="prefix" select="substring($given-name,1,1)"/>
  <xsl:variable name="object-name" select="concat($prefix,$surname)"/
>

  <!-- then see if name already exists in NDS -->
  <xsl:variable name="exists">
    <xsl:call-template name="query-object-name">
      <xsl:with-param name="object-name" select="$object-name"/>
    </xsl:call-template>
  </xsl:variable>

  <!-- if exists, then try 1st fallback, else return result -->
  <xsl:choose>
    <xsl:when test="$exists != ''">
      <xsl:call-template name="create-object-name-2"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="$object-name"/>
    </xsl:otherwise>
  </xsl:choose>

</xsl:template>

<!-- create-object-name-2 is the first fallback if the name created by
-->
<!-- create-object-name already exists
-->
<xsl:template name="create-object-name-2">

  <!-- first try is first name followed by surname -->
  <xsl:variable name="given-name" select="add-attr[@attr-name='Given

```

```

Name']/value"/>
  <xsl:variable name="surname" select="add-attr[@attr-
name='Surname']/value"/>
  <xsl:variable name="object-name" select="concat($given-
name,$surname)"/>

  <!-- then see if name already exists in NDS -->
  <xsl:variable name="exists">
    <xsl:call-template name="query-object-name">
      <xsl:with-param name="object-name" select="$object-name"/>
    </xsl:call-template>
  </xsl:variable>

  <!-- if exists, then try last fallback, else return result -->
  <xsl:choose>
    <xsl:when test="$exists != ''">
      <xsl:call-template name="create-object-name-fallback"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="$object-name"/>
    </xsl:otherwise>
  </xsl:choose>

</xsl:template>

<!-- create-object-name-fallback recursively tries a name created by
-->
<!-- concatenating the surname and a count until NDS doesn't find
-->
<!-- the name. There is a danger of infinite recursion, but only if
-->
<!-- there is a bug in NDS
-->
<xsl:template name="create-object-name-fallback">
  <xsl:param name="count" select="1"/>

  <!-- construct the a name based on the surname and a count -->
  <xsl:variable name="surname" select="add-attr[@attr-
name='Surname']/value"/>
  <xsl:variable name="object-name" select="concat($surname,'-
',$count)"/>

  <!-- see if it exists in NDS -->
  <xsl:variable name="exists">
    <xsl:call-template name="query-object-name">
      <xsl:with-param name="object-name" select="$object-name"/>
    </xsl:call-template>
  </xsl:variable>

  <!-- if exists, then try again recursively, else return result -->
  <xsl:choose>
    <xsl:when test="$exists != ''">
      <xsl:call-template name="create-object-name-fallback">
        <xsl:with-param name="count" select="$count + 1"/>
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="$object-name"/>
    </xsl:otherwise>
  </xsl:choose>

```

```

        </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
        <xsl:value-of select="$object-name"/>
    </xsl:otherwise>
</xsl:choose>

</xsl:template>

<!-- query object name queries NDS for the passed object-name. Ideally,
this would -->
<!-- not depend on "CN": to do this, add another parameter that is the
name of the -->
<!-- naming attribute.
-->
<xsl:template name="query-object-name">
    <xsl:param name="object-name"/>

    <!-- build an xds query as a result tree fragment -->
    <xsl:variable name="query">
        <nds ndsversion="8.5" dtdversion="1.0">
            <input>
                <query>
                    <search-class class-name="{ancestor-or-self::add/@class-
name}"/>
                    <!-- NOTE: depends on CN being the naming attribute -->
                    <search-attr attr-name="CN">
                        <value><xsl:value-of select="$object-name"/></value>
                    </search-attr>
                    <!-- put an empty read attribute in so that we don't get
the whole object back -->
                    <read-attr/>
                </query>
            </input>
        </nds>
    </xsl:variable>

    <!-- query NDS -->
    <xsl:variable name="result"
select="query:query($destQueryProcessor,$query)"/>

    <!-- return an empty or non-empty result tree fragment depending on
result of query -->
    <xsl:value-of select="$result//instance"/>
</xsl:template>

<!-- create an initial password -->
<xsl:template name="create-password">
    <password>
        <xsl:value-of select="concat(add-attr[@attr-name='Surname']/
value,'-',add-attr[@attr-name='CN']/value)"/>
    </password>
</xsl:template>

```

```
<!-- identity transform for everything we don't want to mess with -->
<xsl:template match="@*|node()">
  <xsl:copy>
    <xsl:apply-templates select="@*|node()"/>
  </xsl:copy>
</xsl:template>

</xsl:transform>
```


管理過濾器

「過濾器」編輯器可讓您管理過濾器。在「過濾器」編輯器中，您可定義「發行者」和「訂閱者」通道處理每個類別和屬性的方式。

本節涵蓋下列與過濾器相關的主題：

- ◆ 「Designer 中的過濾器任務」，第 365 頁
- ◆ 「iManager 中的過濾器任務」，第 385 頁

6.1 Designer 中的過濾器任務


本節包含在 Designer 中執行常見的過濾器相關任務的指示：

- ◆ 「存取過濾器編輯器」，第 365 頁
- ◆ 「編輯過濾器」，第 368 頁
- ◆ 「測試過濾器」，第 372 頁
- ◆ 「檢視過濾器的 XML 來源」，第 378 頁
- ◆ 「其他過濾器選項」，第 383 頁

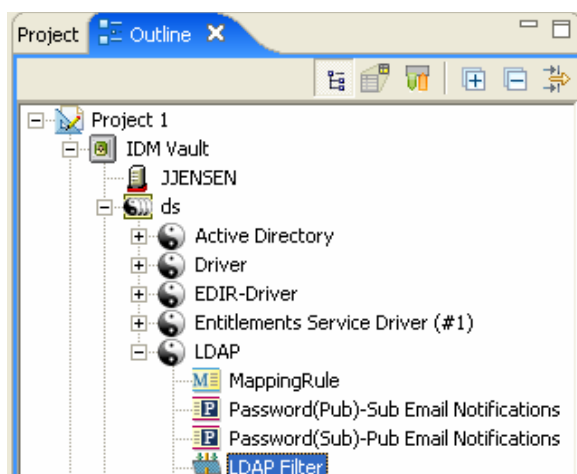
6.1.1 存取過濾器編輯器

「過濾器」編輯器可讓您編輯過濾器。有三種存取「過濾器」編輯器的不同方法：透過模型大綱、透過規則流程，以及透過「規則集」檢視窗。

模型大綱檢視窗

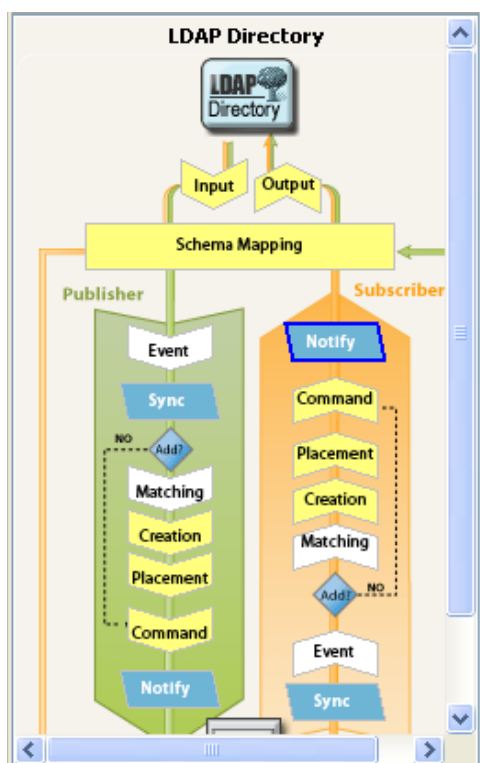
- 1 在開啓的專案中，按一下「大綱」索引標籤。
 - 2 按一下「顯示模型大綱」圖示。
 - 3 選取您要管理其過濾器的驅動程式，然後按一下右邊的加號。
 - 4 連接兩下「過濾器」圖示，並啓動「過濾器」編輯器。
- 或

按一下滑鼠右鍵，並選取「編輯」。



規則流程檢視窗

- 1 在開啓的專案中，按一下「大綱」索引標籤。
- 2 選取「顯示規則流程」圖示。
- 3 連接兩下「同步化」圖示或「通知」圖示，啓動「過濾器」編輯器。

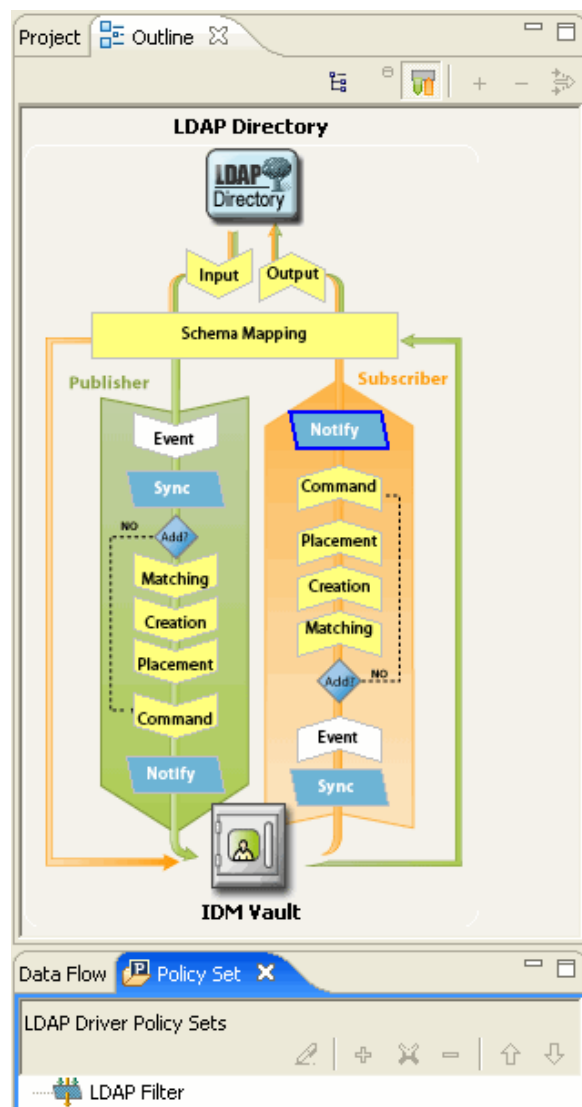


- 4 當「規則流程」下的「規則集管理員」中出現過濾器時，連接兩下過濾器以啓動「過濾器」編輯器。
或

按一下滑鼠右鍵，並選取「編輯規則 > 過濾器」。

規則集檢視窗

1 連按兩下過濾器規則。



鍵盤支援

表格 6-1 過濾器編輯器的鍵盤支援

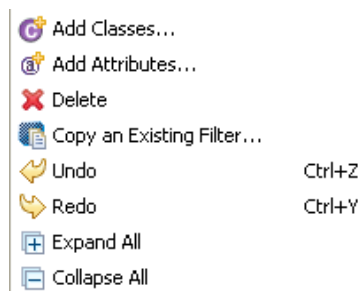
動作	描述
向上鍵	在「過濾器」編輯器中，將游標上移。
向下鍵	在「過濾器」編輯器中，將游標下移。
向左鍵	折疊顯示的資訊

動作	描述
向右鍵	展開顯示的資訊。
Insert	新增類別。
Ctrl+Insert	新增屬性。
Delete	刪除選定項目。
Enter	存取編輯模式。第二次按下 Enter 時可認可變更。
Esc	離開編輯模式。

6.1.2 編輯過濾器

「過濾器」編輯器可讓您建立及編輯過濾器。若要顯示快顯功能表，請以滑鼠右鍵按一下項目。

特性 6-1 過濾器選項




- ◆ 「移除或新增類別和屬性」，第 368 頁
- ◆ 「修改多個屬性」，第 369 頁
- ◆ 「複製現有的過濾器」，第 369 頁
- ◆ 「設定屬性的預設值」，第 369 頁
- ◆ 「變更過濾器設定」，第 370 頁

移除或新增類別和屬性

藉由移除或新增類別和屬性，您可決定在已連接資料儲存和 Identity Vault 之間進行同步化的物件。

移除類別或屬性


如果不要某個類別或屬性進行同步化，最佳的作法是將過濾器中的類別或屬性完全移除。有兩種方法可以新增或移除過濾器中的屬性和類別：

- ◆ 以滑鼠右鍵按一下要移除的類別或屬性，然後選取「刪除」。
- ◆ 選取要移除的類別或屬性，然後按一下右上角的「刪除」圖示 .

新增類別

- 1 在「過濾器」編輯器中按一下滑鼠右鍵，然後按一下「新增類別」。

或


按一下右上角的「新增類別」圖示 

- 1 瀏覽至要新增的類別並加以選取，然後按一下「確定」。
- 2 變更選項以同步化資訊。
- 3 若要儲存變更，請按一下「檔案 > 儲存」。

新增屬性

- 1 在「過濾器」編輯器中按一下滑鼠右鍵，然後按一下「新增屬性」。

或

按一下右上角的屬性圖示 


- 2 瀏覽至要新增的屬性並加以選取，然後按一下「確定」。
- 3 變更選項以同步化資訊。
- 4 若要儲存變更，請按一下「檔案 > 儲存」。

修改多個屬性

「過濾器」編輯器可讓您一次修改多個屬性。按下 Ctrl 鍵並選取多個屬性；選項變更時，所有選定屬性都會隨之變更。

複製現有的過濾器

您可以從另一個驅動程式複製現有的過濾器，並在您目前使用的驅動程式中加以使用。

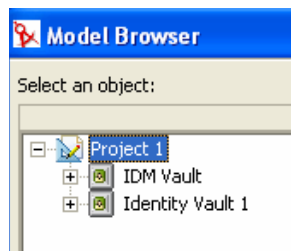
- 1 按一下「複製現有的過濾器」圖示 

或

在「過濾器」編輯器中按一下滑鼠右鍵，然後按一下「複製現有的過濾器」。


- 2 瀏覽至要複製的過濾器物件並加以選取，然後按一下「確定」。

如果專案中有多個 Identity Vault，您可以從其他 Identity Vault 複製過濾器。瀏覽至其他物件並加以選取時，您可以瀏覽至其他 Identity Vault，並且使用儲存於該處的過濾器。



設定屬性的預設值

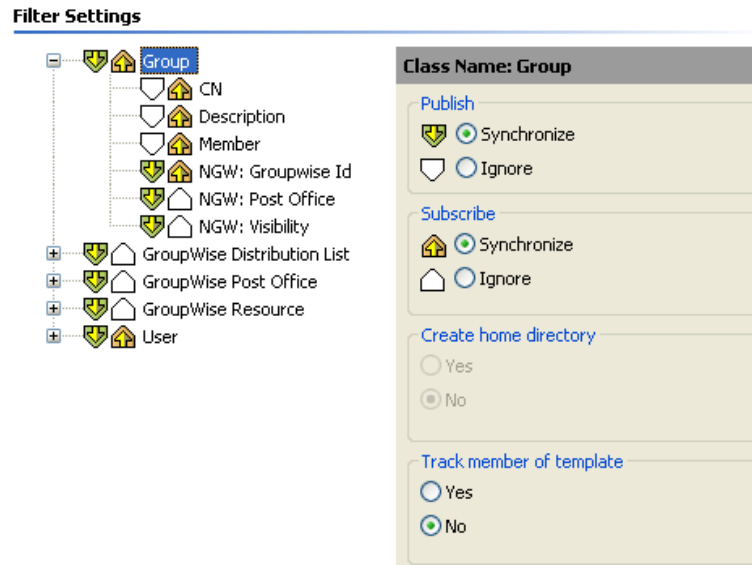
新的屬性新增至過濾器時，您可以定義這些屬性的預設值。

- 1 按一下右上角的「設定新屬性的預設值」圖示 
- 2 選取要讓新屬性擁有的選項，然後按一下「確定」。

變更過濾器設定

「過濾器」編輯器可讓您選擇變更在 Identity Vault 和已連接系統之間同步化資訊的方式。過濾器的類別和屬性設定都不一樣。

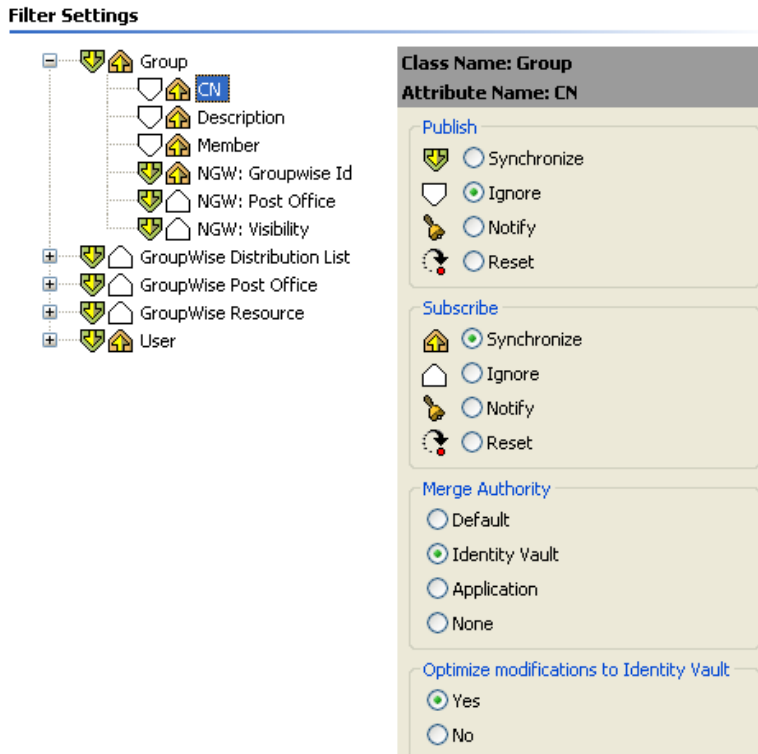
- 1 在「過濾器」編輯器中選取類別。



- 2 變更選定類別的過濾器設定。

選項	定義
發行者	<ul style="list-style-type: none">◆ 同步化：可讓類別從已連接系統同步化到 Identity Vault 中。◆ 忽略：不從已連接系統將類別同步化到 Identity Vault 中。
訂閱者	<ul style="list-style-type: none">◆ 同步化：可讓類別從 Identity Vault 同步化到已連接系統中。◆ 忽略：不從 Identity Vault 將類別同步化到已連接系統中。
建立主目錄	<ul style="list-style-type: none">◆ 是：自動建立主目錄。◆ 否：不建立主目錄。
追蹤範本的成員	<ul style="list-style-type: none">◆ 是：決定「發行者」通道從範本建立物件時，是否保留 Member of Template 屬性。◆ 否：不追蹤 Member of Template 屬性。

3 選取屬性。



4 變更選定屬性的過濾器設定。


選項	定義
發行者	<ul style="list-style-type: none"> ◆ 同步化：報告並自動同步化對於此物件的變更。 ◆ 忽略：不報告且不自動同步化對於此物件的變更。 ◆ 通知：報告對於此物件的變更，但不自動同步化。 ◆ 重設：將物件值重設為相反通道所指定的值。(您可以在「發行者」通道或「訂閱者」通道上設定此值，但不能同時在兩個通道上設定。)
訂閱者	<ul style="list-style-type: none"> ◆ 同步化：報告並自動同步化對於此物件的變更。 ◆ 忽略：不報告且不自動同步化對於此物件的變更。 ◆ 通知：報告對於此物件的變更，但不自動同步化。 ◆ 重設：將物件值重設為相反通道所指定的值。(您可以在「發行者」通道或「訂閱者」通道上設定此值，但不能同時在兩個通道上設定。)

選項	定義
合併權限	<ul style="list-style-type: none"> ◆ 預設行爲：如果屬性不是在任一通道上同步化，則不會發生合併。如果屬性在一個通道上同步化，而不在另一個通道上同步化，則該通道之目的地上的所有現有值便會移除，並取代為來自該通道之來源的值。如果來源有多個值，且目的地只能容納單一值，則在目的地端只會使用其中一個值。 如果屬性在兩個通道上都同步化，而且兩端都只能容納單一值，則已連接應用程式會取得 Identity Vault 值，除非 Identity Vault 中沒有值。如果是這種情況，Identity Vault 會從已連接應用程式取得值（如果有的話）。 如果屬性在兩個通道上都同步化，而且只有一端可以容納多個值，則單一值端的值會新增至多值端（如果該值尚未在多值端）。如果單一值端沒有值，您可以選擇值並將其新增至單一值端。 此固定為有效行爲。 ◆ Identity Vault：如果屬性在「訂閱者」通道而不在「發行者」通道上同步化，則進行和預設行爲相同的行爲。 這是在「訂閱者」通道上進行同步化時的有效行爲。 ◆ 應用程式：如果屬性在「發行者」通道而不在「訂閱者」通道上同步化，則進行和預設行爲相同的行爲。 這是在「發行者」通道上進行同步化時的有效行爲。 ◆ 無：不發生合併（無視同步化）。
對 Identity Manager 進行最佳化修改	<ul style="list-style-type: none"> ◆ 是：對於此屬性的變更會在「發行者」通道上經過檢查，以便決定在 Identity Vault 中進行的最少變更。 ◆ 否：不檢查變更。

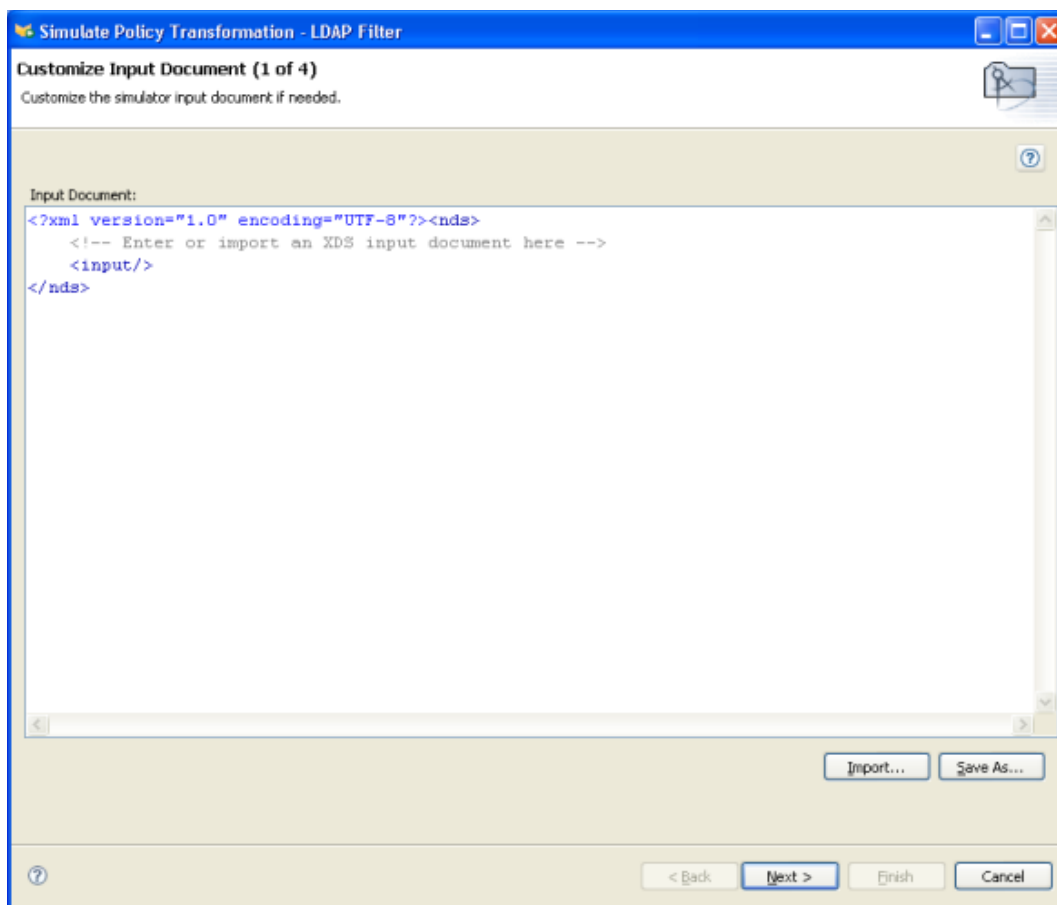
5 按一下「儲存」圖示  儲存變更。

6.1.3 測試過濾器

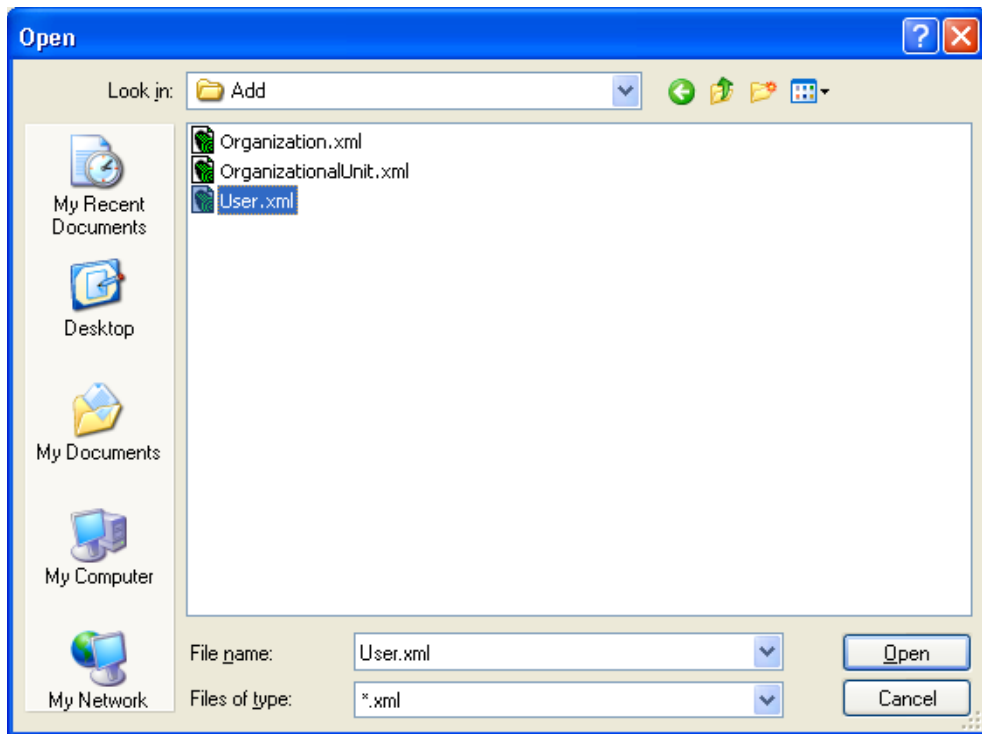
Designer 隨附一套稱為「規則模擬器」的工具，可讓您測試規則而不在生產環境中加以實作。您可以透過「過濾器」編輯器啟動「規則模擬器」，以便在修改規則後加以測試。

1 按一下工具列中的「啟動規則模擬器」圖示 .

2 選取「輸入」以瀏覽至模擬事件的檔案。

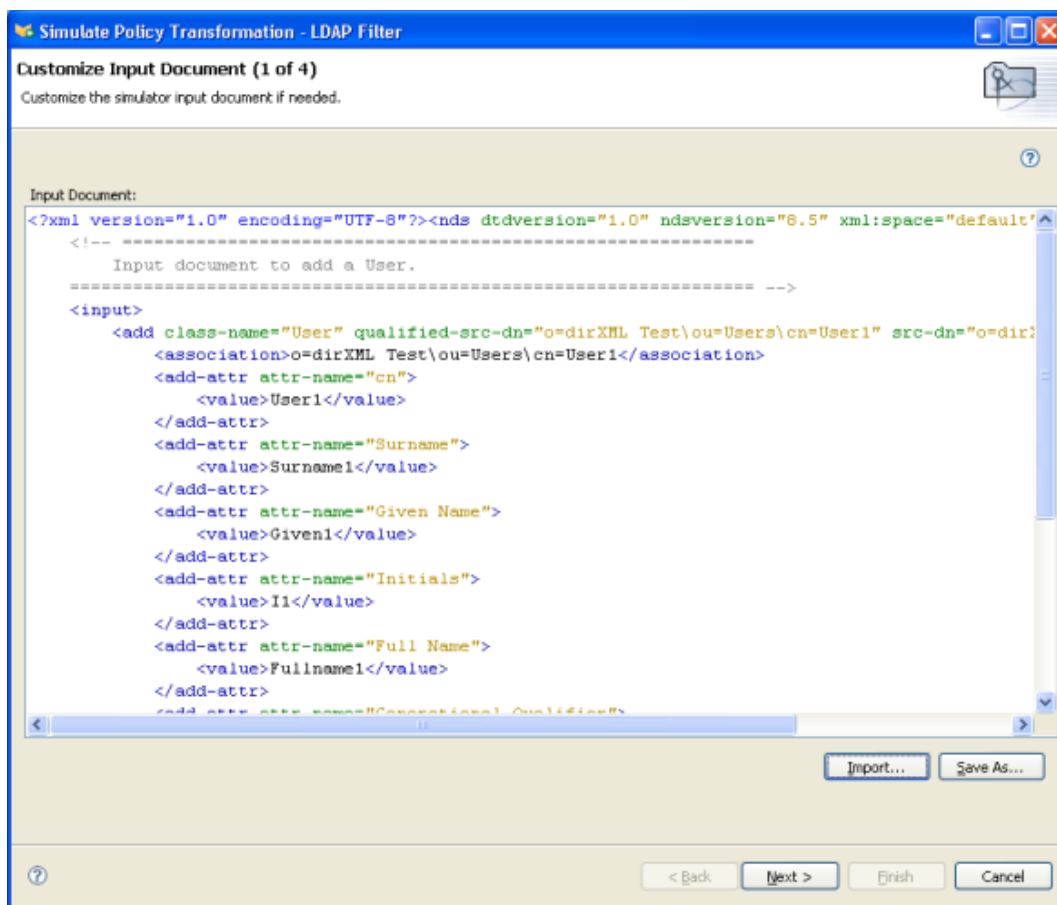


- 3 選取檔案，然後按一下「開啓」。此範例使用 `com.novell.designer.idm.policy\simulation\add\User.xml` 檔案，該檔案會針對「使用者」物件模擬「新增」事件。



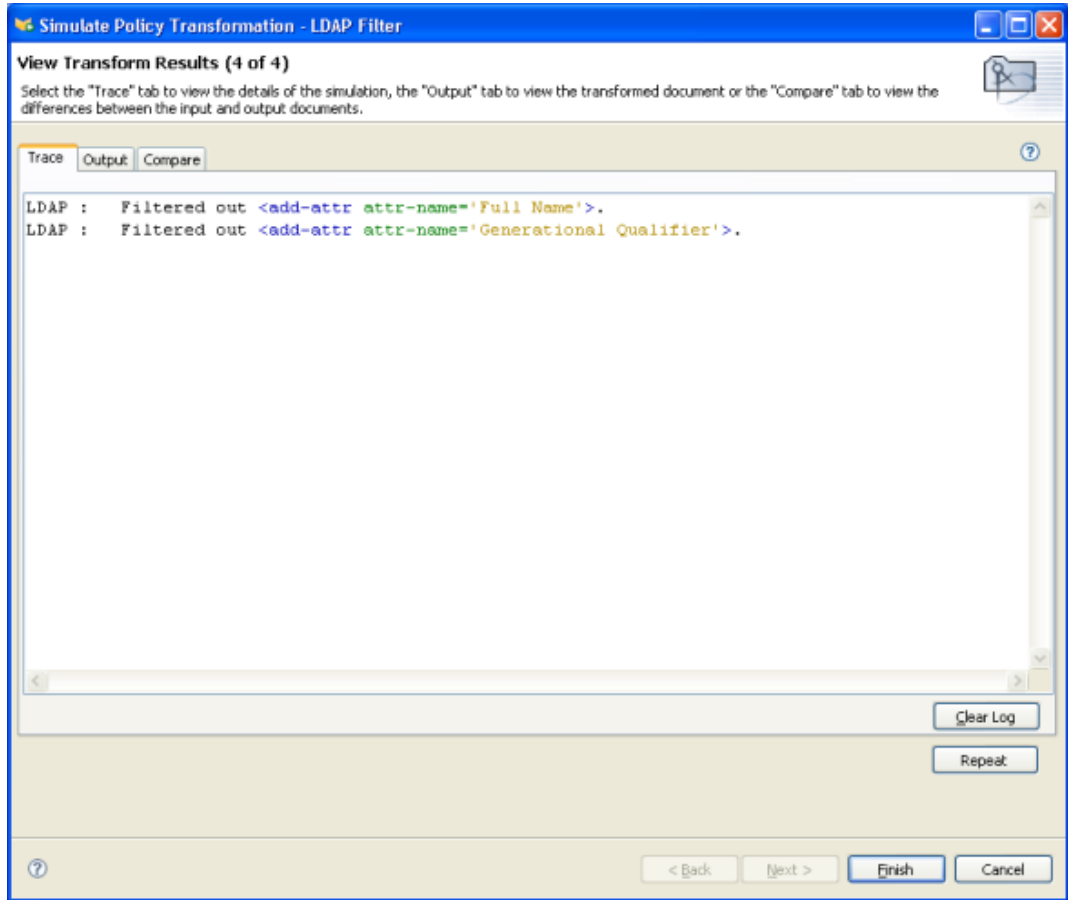
「規則模擬器」會顯示使用者「新增」事件的輸入文件。

4 按「下一步」開始模擬。

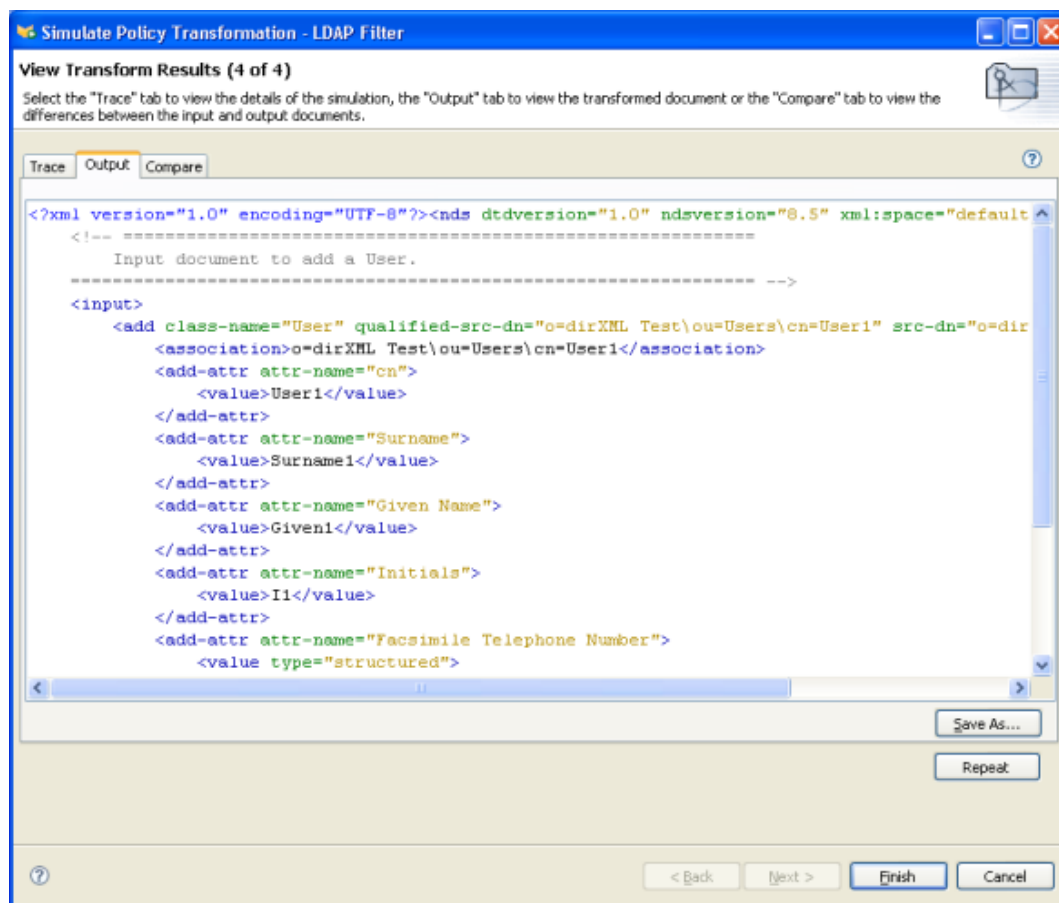


「規則模擬器」會顯示「新增」事件的記錄、輸出文件，以及「輸入」文件與產生之「輸出」文件的比較。

- 5 選取「追蹤」索引標籤以顯示「新增」事件的結果，您會在 DSTRACE 中看到這些結果。

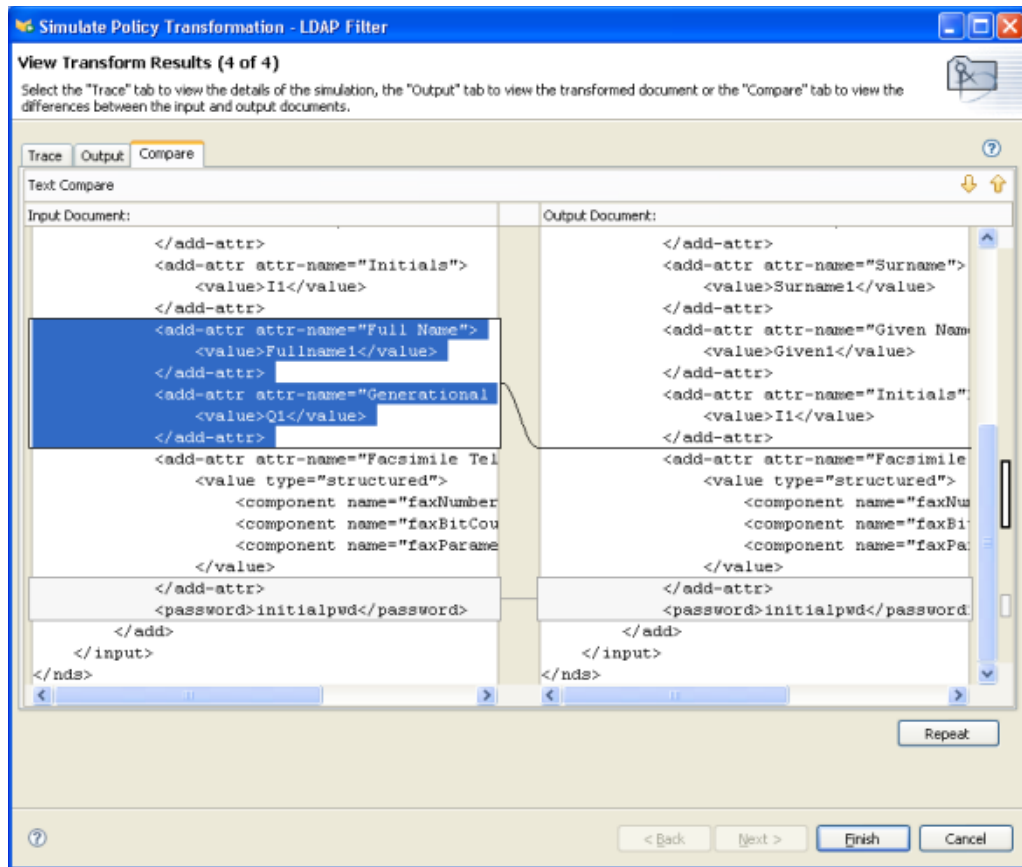


- 6 選取「輸出」索引標籤，以查看針對輸入文件執行過濾器時產生的輸出文件。輸入文件即為使用者「新增」事件。



您可以編輯輸入和輸出文件。如果要保留變更，請按一下「另存新檔」。

7 選取「比較」索引標籤，以比較輸入文件的文字與產生的輸出文件。



8 按一下「重複」以選取不同的輸入文件，並查看該事件的結果。

9 當您完成過濾器的測試時，請按一下「完成」，關閉「規則模擬器」。

6.1.4 檢視過濾器的 XML 來源

Designer 可讓您藉由使用 XML 編輯器或文字編輯器來檢視、編輯及驗證 XML。

- ◆ 「[檢視 XML 來源](#)」，第 378 頁
- ◆ 「[編輯 XML 來源](#)」，第 381 頁
- ◆ 「[驗證 XML 來源](#)」，第 383 頁

檢視 XML 來源

您可以 XML 或以 XML 樹狀結構格式檢視「XML 來源」。

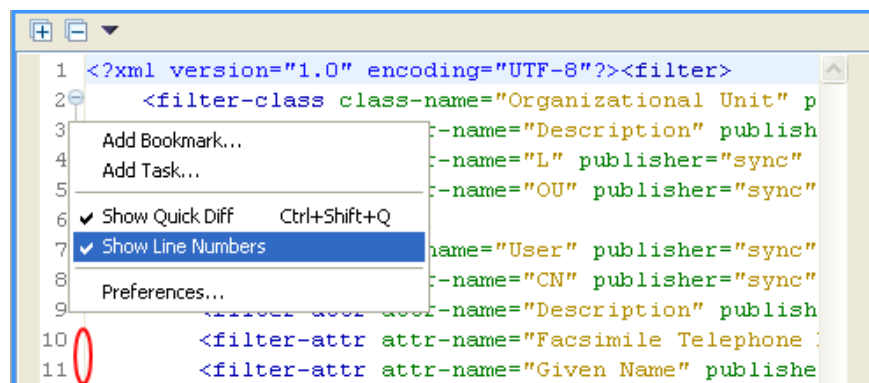
若要開啓「XML 來源」檢視窗，請執行下列動作：

- 1 按一下位於「過濾器」編輯器之工作空間底部的「XML 來源」。



XML 編輯器會顯示行號。若要查看行號，請以滑鼠右鍵按一下左邊界，然後選取「顯示行號」。

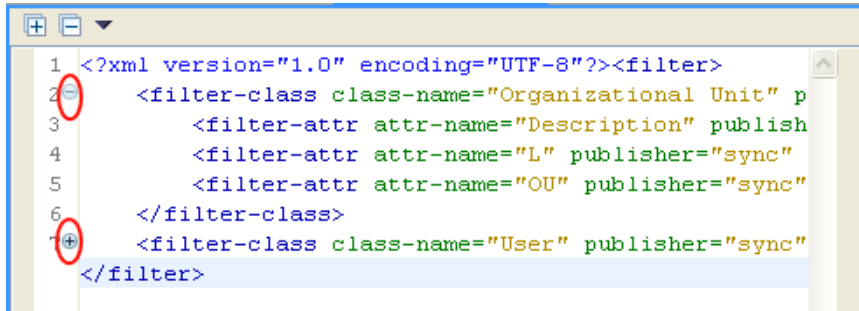
特性 6-2 過濾器的顯示行號



XML 編輯器會依功能展開或折疊 XML。如果有些功能包含大量的 XML，您可以按一下位於左上角的減號圖示，以折疊 XML。若要展開所有的 XML 功能，請按一下位於左上角的加號圖示。

每個元素的左邊界皆擁有其各自的加號或減號圖示。

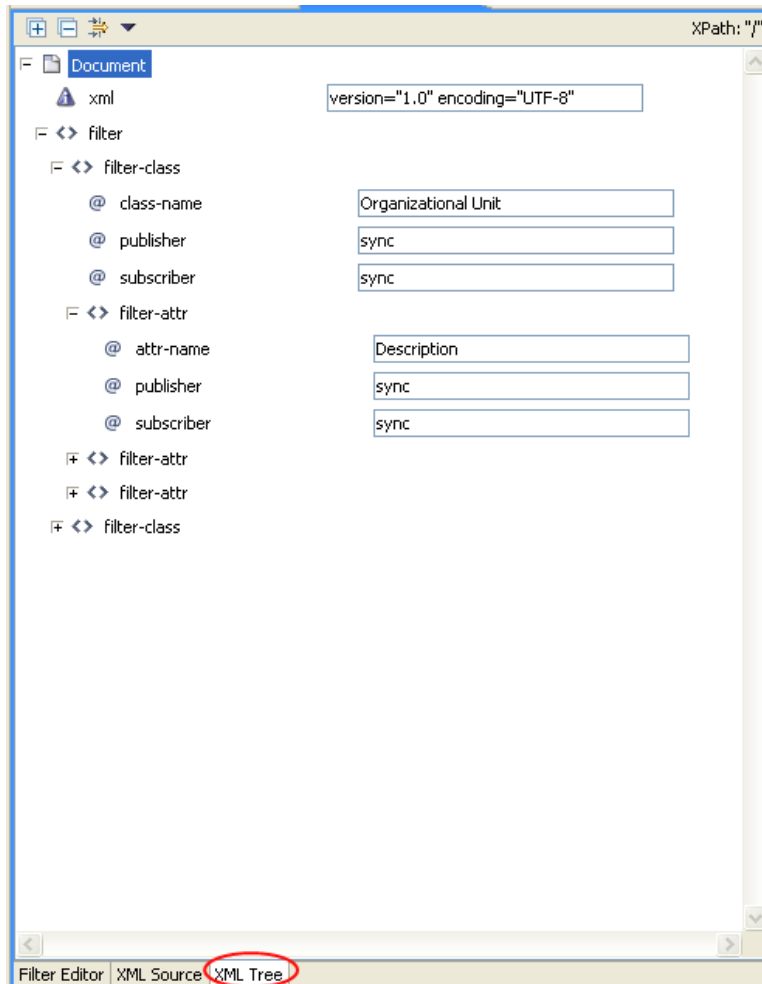
特性 6-3 過濾器的 XML 加號或減號



```
1 <?xml version="1.0" encoding="UTF-8"?><filter>
2   <filter-class class-name="Organizational Unit" p
3     <filter-attr attr-name="Description" publish
4     <filter-attr attr-name="L" publisher="sync"
5     <filter-attr attr-name="OU" publisher="sync"
6   </filter-class>
7   <filter-class class-name="User" publisher="sync"
  </filter>
```

若要以樹狀結構格式檢視 XML，請執行下列動作：

- 1 按一下位於「過濾器」編輯器之工作空間底部的「XML 樹狀結構」。

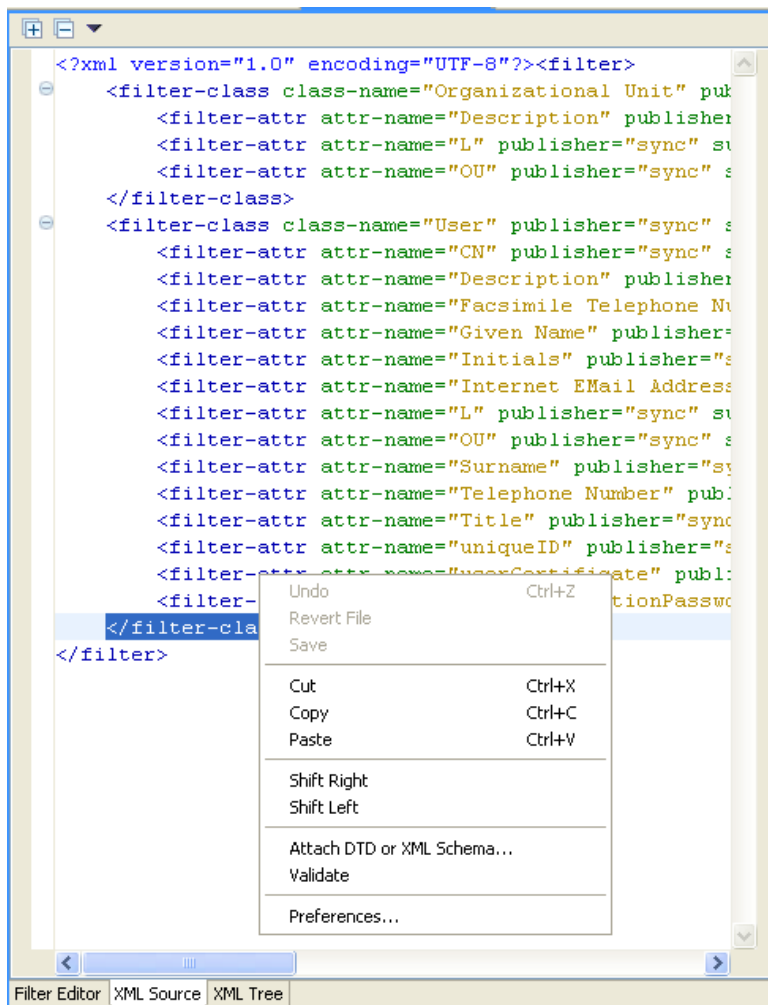


若要查看整個樹狀結構檢視窗，請展開列出的每個項目。

編輯 XML 來源

您可以透過 XML 編輯器編輯 XML。您也可以透過使用者圖形介面 (Graphic User Interface, GUI) 在此處進行變更。

特性 **6-4** 編輯過濾器的 XML 來源



載入的預設編輯器與 .xml 檔案類型相關聯。若找不到預設編輯器，則會載入系統文字編輯器。「XML 來源」檢視窗功能是以進行載入的編輯器為基礎。

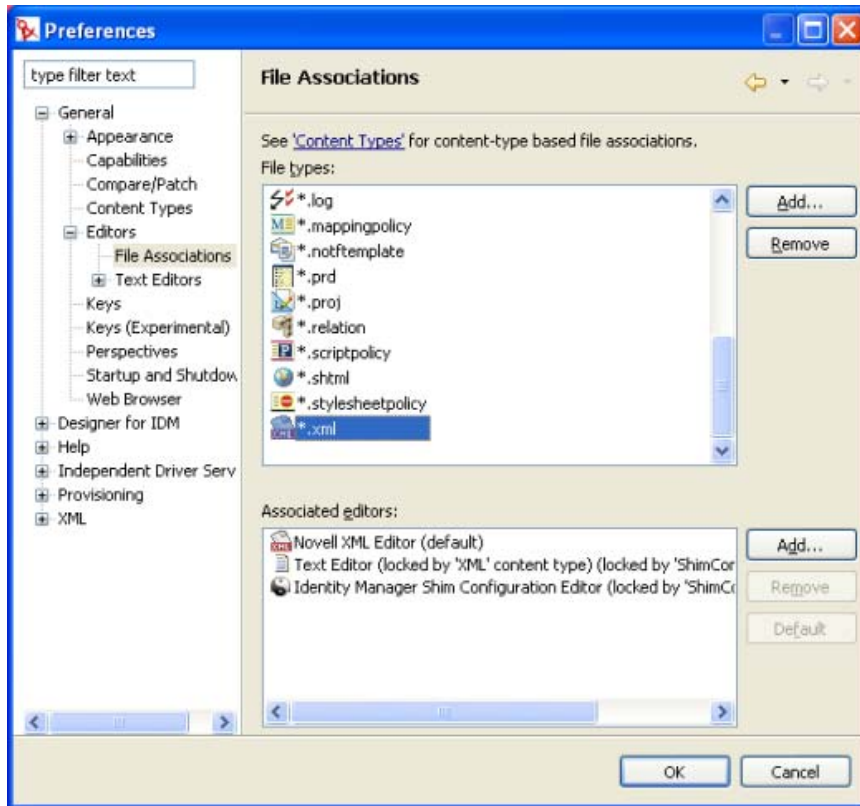
按一下滑鼠右鍵，可顯示 XML 編輯器包含的功能清單。

- ◆ 復原：復原上一個動作。
- ◆ 回復檔案：將檔案回復到上一次儲存的版本。
- ◆ 儲存：儲存檔案。
- ◆ 剪下：剪下選定資訊。
- ◆ 複製：將選定資訊複製到「剪貼簿」。
- ◆ 貼上：將資訊貼至文件。

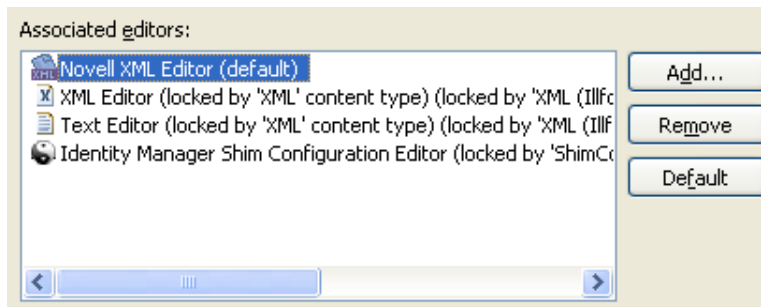
- ◆ 右移：向右縮排。
- ◆ 左移：向左縮排。
- ◆ 附加 DTD 或 XML 綱要：附加用於驗證規則的 DTD 或 XML 綱要檔案。
- ◆ 驗證：驗證 XML 程式碼。
- ◆ 優先設定：設定 XML 編輯器的優先設定。

若要為您的「XML 來源」檢視窗選擇不同的 XML 編輯器，請執行下列動作：

- 1 按一下「主功能表」中的「視窗 > 優先設定」。
- 2 按一下「一般 > 編輯器 > 檔案關聯」。
- 3 選取檔案類型清單中的「*.xml」。



- 4 從「關聯的編輯器」選取所需的編輯器（如「Novell XML Editor」）。（若您要使用的編輯器未出現在清單中，可以按一下「新增」，然後將其新增至清單中）。



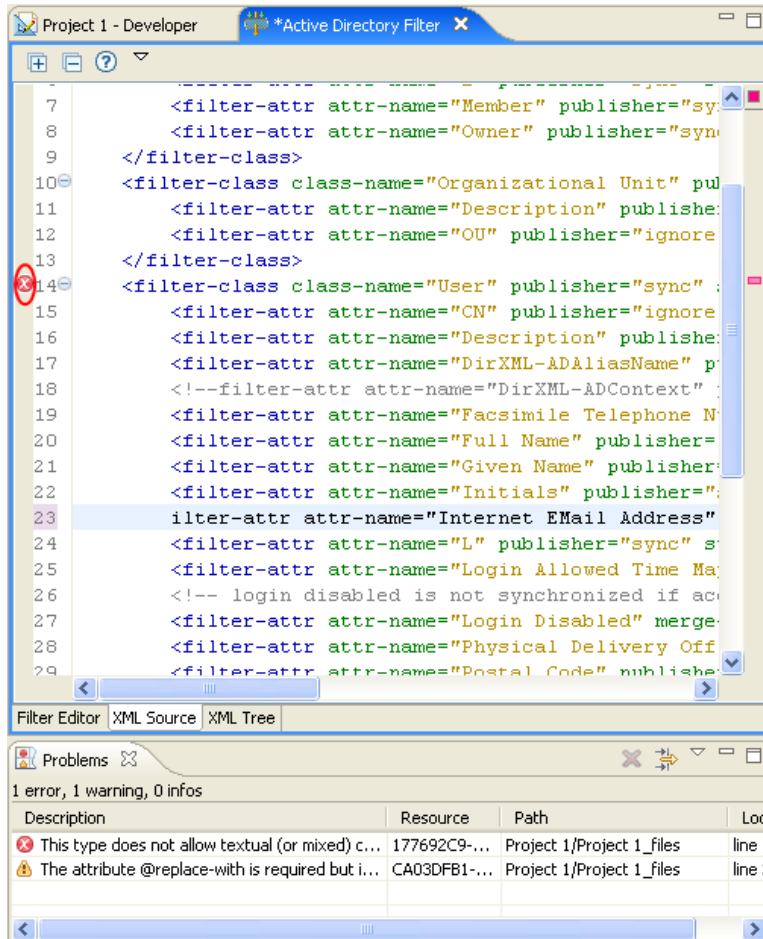
5 按一下「確定」。

6 關閉並重新開啓「過濾器」編輯器。預設編輯器應在「XML 來源」檢視窗中載入。

驗證 XML 來源

XML 編輯器會驗證 XML 程式碼。按一下滑鼠右鍵，然後選取「驗證」。若出現錯誤，則在發生錯誤的該行上會顯示一個紅色的 x。視窗底端的說明提供了該問題的相關資訊。

特性 6-5 驗證過濾器



在此範例中，<filter-attr> 的開頭標籤和第一個字母不見了。

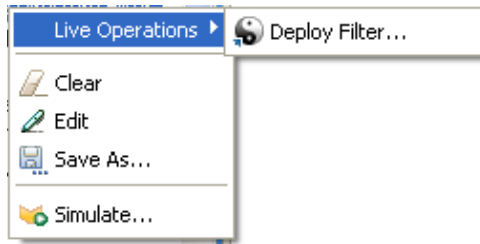
6.1.5 其他過濾器選項

以滑鼠右鍵按一下過濾物件時，「大綱」檢視窗、「規則流程」檢視窗和「規則集」檢視窗中會出現多個選項。

- ◆ 「大綱檢視窗的其他選項」，第 384 頁
- ◆ 「規則流程檢視窗的其他選項」，第 384 頁
- ◆ 「規則集檢視窗的其他選項」，第 384 頁

大綱檢視窗的其他選項

- 1 在「大綱」檢視窗中，以滑鼠右鍵按一下過濾物件。



- ◆ 即時操作 > 部署過濾器：將過濾器部署至 Identity Vault。
- ◆ 清除：刪除過濾器規則中的所有內容，但保留物件。
- ◆ 編輯：啟動「過濾器」編輯器。如需相關資訊，請參閱「[編輯過濾器](#)」，第 368 頁。
- ◆ 另存新檔：將「過濾器」儲存為 .xml 檔案。
- ◆ 模擬：啟動「規則模擬器」。如需相關資訊，請參閱「[測試過濾器](#)」，第 372 頁。

規則流程檢視窗的其他選項

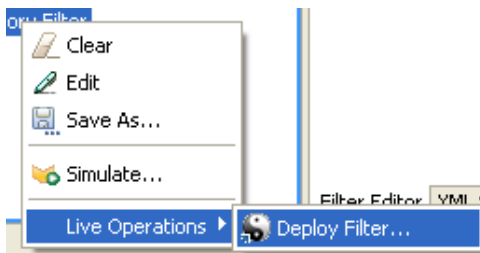
- 1 在「規則流程」檢視窗中，以滑鼠右鍵按一下過濾物件。



- ◆ 編輯規則 > 過濾器：啟動「過濾器」編輯器。如需相關資訊，請參閱「[編輯過濾器](#)」，第 368 頁。
- ◆ 模擬：啟動「規則模擬器」。如需相關資訊，請參閱「[測試過濾器](#)」，第 372 頁。

規則集檢視窗的其他選項

- 1 在「規則集」檢視窗中，以滑鼠右鍵按一下過濾物件。



- ◆ 清除：刪除過濾器規則中的所有內容，但保留物件。
- ◆ 編輯：啟動「過濾器」編輯器。如需相關資訊，請參閱「[編輯過濾器](#)」，第 385 頁。
- ◆ 儲存：將過濾器儲存為 XML 檔案。
- ◆ 模擬：啟動「規則模擬器」。如需相關資訊，請參閱「[測試過濾器](#)」，第 372 頁。

- ◆ 即時操作 > 部署過濾器：可讓您將過濾器部署至 Identity Vault。

6.2 iManager 中的過濾器任務

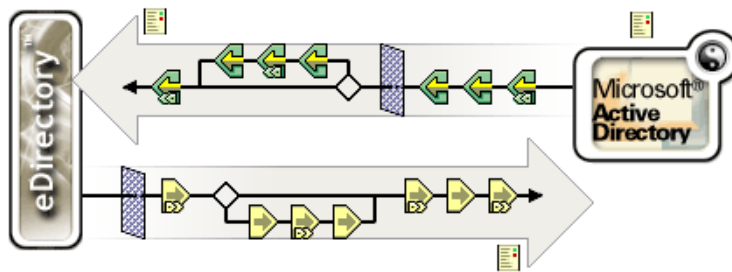
本節包含在 iManager 中執行常見的過濾器相關任務的指示：

- ◆ 「存取過濾器」，第 385 頁
- ◆ 「編輯過濾器」，第 385 頁

6.2.1 存取過濾器

- 1 在 iManager 中展開「Identity Manager」角色，然後按一下「Identity Manager 概觀」。
- 2 選取「搜尋整個網路樹」或「在容器中搜尋」，然後按一下「搜尋」。
- 3 按一下要存取其過濾器的驅動程式。此時會開啓「Identity Manager 驅動程式概觀」：

特性 6-6 驅動程式概觀



- 4 按一下「發行者」或「訂閱者」通道上的「過濾器」圖示。該圖示為相同物件。



6.2.2 編輯過濾器

「過濾器」編輯器可讓您選擇編輯在 Identity Vault 和已連接系統之間同步化資訊的方式。以下是編輯過濾器時最常見的任務清單：

- ◆ 「移除過濾器中的類別或屬性」，第 385 頁
- ◆ 「新增類別」，第 386 頁
- ◆ 「新增屬性」，第 386 頁
- ◆ 「複製過濾器」，第 386 頁
- ◆ 「設定範本」，第 386 頁
- ◆ 「變更過濾器設定」，第 386 頁

移除過濾器中的類別或屬性

- 1 選取類別或屬性，然後按一下「刪除」。

新增類別

- 1 按一下「新增類別」。
- 2 變更選項以同步化資訊。
- 3 按一下「套用」。

新增屬性

- 1 按一下「新增屬性」。
- 2 變更選項以同步化資訊。
- 3 按一下「套用」。

複製過濾器

可讓您將過濾器從現有驅動程式複製到目前使用的驅動程式中。

- 1 按一下「過濾器的複製來源」。
- 2 瀏覽至要從其中複製過濾器的驅動程式，然後按一下「確定」。

設定範本

可讓您為新增至過濾器的屬性設定預設值。

- 1 按一下「設定範本」。
- 2 選取要讓新屬性具有的選項，然後按一下「確定」。

屬性建立後，您可以變更其值。

變更過濾器設定

「過濾器」編輯器可讓您選擇變更在 Identity Vault 和已連接系統之間同步化資訊的方式。過濾器的類別和屬性設定都不一樣。

- 1 在「過濾器」編輯器中選取類別。
- 2 變更選定類別的過濾器設定。

選項	定義
發行者	<ul style="list-style-type: none">◆ 同步化：可讓類別從已連接系統同步化到 Identity Vault 中。◆ 忽略：不從已連接系統將類別同步化到 Identity Vault 中。
訂閱者	<ul style="list-style-type: none">◆ 同步化：可讓類別從 Identity Vault 同步化到已連接系統中。◆ 忽略：不從 Identity Vault 將類別同步化到已連接系統中。
建立主目錄	<ul style="list-style-type: none">◆ 是：自動建立主目錄。◆ 否：不建立主目錄。
追蹤範本的成員	<ul style="list-style-type: none">◆ 是：決定「發行者」通道從範本建立物件時，是否保留 Member of Template 屬性。◆ 否：不追蹤 Member of Template 屬性。

- 3 選取屬性。

4 變更選定屬性的過濾器設定。

選項	定義
發行者	<ul style="list-style-type: none">◆ 同步化：報告並自動同步化對於此物件的變更。◆ 忽略：不報告且不自動同步化對於此物件的變更。◆ 通知：報告對於此物件的變更，但不自動同步化。◆ 重設：將物件值重設為相反通道所指定的值。(您可以在「發行者」通道或「訂閱者」通道上設定此值，但不能同時在兩個通道上設定。)
訂閱者	<ul style="list-style-type: none">◆ 同步化：報告並自動同步化對於此物件的變更。◆ 忽略：不報告且不自動同步化對於此物件的變更。◆ 通知：報告對於此物件的變更，但不自動同步化。◆ 重設：將物件值重設為相反通道所指定的值。(您可以在「發行者」通道或「訂閱者」通道上設定此值，但不能同時在兩個通道上設定。)
合併權限	<ul style="list-style-type: none">◆ 預設行爲：如果屬性不是在任一通道上同步化，則不會發生合併。 如果屬性在一個通道上同步化，而不在另一個通道上同步化，則該通道之目的地上的所有現有值便會移除，並取代為來自該通道之來源的值。如果來源有多個值，且目的地只能容納單一值，則在目的地端只會使用其中一個值。 如果屬性在兩個通道上都同步化，而且兩端都只能容納單一值，則已連接應用程式會取得 Identity Vault 值，除非 Identity Vault 中沒有值。如果是這種情況，Identity Vault 會從已連接應用程式取得值 (如果有的話)。 如果屬性在兩個通道上都同步化，而且只有一端可以容納多個值，則單一值端的值會新增至多值端 (如果該值尚未在多值端)。如果單一值端沒有值，您可以選擇值並將其新增至單一值端。 此固定為有效行爲。◆ Identity Vault：如果屬性在「訂閱者」通道而不在「發行者」通道上同步化，則進行和預設行爲相同的行爲。 這是在「訂閱者」通道上進行同步化時的有效行爲。◆ 應用程式：如果屬性在「發行者」通道而不在「訂閱者」通道上同步化，則進行和預設行爲相同的行爲。 這是在「發行者」通道上進行同步化時的有效行爲。◆ 無：不發生合併 (無視同步化)。
對 Identity Manager 進行最佳化修改	<ul style="list-style-type: none">◆ 是：對於此屬性的變更會在「發行者」通道上經過檢查，以便決定在 Identity Vault 中進行的最少變更。◆ 否：不檢查變更。

5 按一下「確定」儲存變更。

管理綱要映射規則

「綱要映射」規則會映射 Identity Vault 名稱空間與應用程式名稱空間之間的類別名稱和屬性名稱。相同的綱要映射規則會套用於雙向。在 Metadirectory 引擎和應用程式 Shim 之間的任一通道上，按任一方向傳遞的所有文件，都會透過「綱要映射」規則傳遞。

每個驅動程式有一個「綱要映射」規則。

本節涵蓋下列與過濾器相關的主題：

- ◆ 「Designer 中的綱要映射規則任務」，第 389 頁
- ◆ 「iManager 中的綱要映射規則任務」，第 411 頁

7.1 Designer 中的綱要映射規則任務

本節包含在 Designer 中執行與「綱要映射」規則相關之常見任務的指示：


- ◆ 「存取綱要映射編輯器」，第 389 頁
- ◆ 「編輯綱要映射規則」，第 393 頁
- ◆ 「測試綱要映射規則」，第 396 頁
- ◆ 「存取綱要映射規則 XML」，第 402 頁
- ◆ 「綱要映射規則的其他選項」，第 408 頁

7.1.1 存取綱要映射編輯器

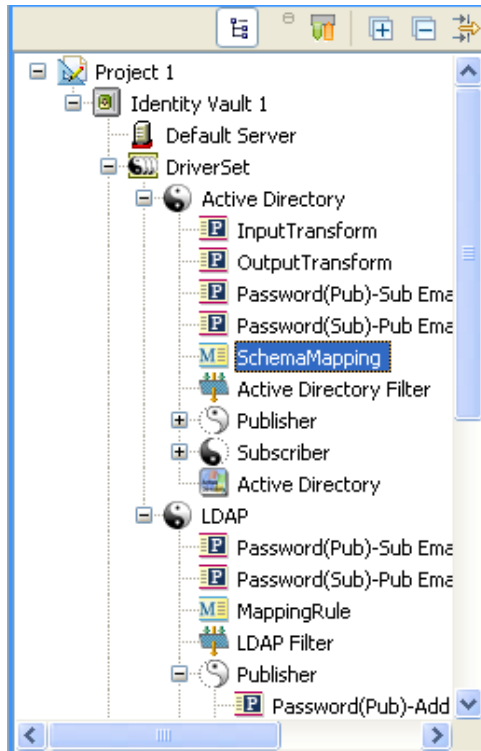
「綱要映射」編輯器可讓您編輯「綱要映射」規則。在 Designer 中存取「綱要映射」編輯器有三種不同的方法：透過「大綱」檢視窗、透過「規則流程」檢視窗，或透過「規則集」檢視窗。

- ◆ 「大綱檢視窗」，第 389 頁
- ◆ 「規則流程檢視窗」，第 390 頁
- ◆ 「規則集檢視窗」，第 391 頁
- ◆ 「鍵盤支援」，第 392 頁

大綱檢視窗

- 1 在開啓的專案中，按一下「大綱」索引標籤。
 - 2 按一下「顯示模型大綱」圖示。
 - 3 選取您要於其上管理綱要映射規則的驅動程式，然後按一下右邊的加號。
 - 4 連按兩下「綱要映射」圖示，以啓動「綱要映射」編輯器。
- 或

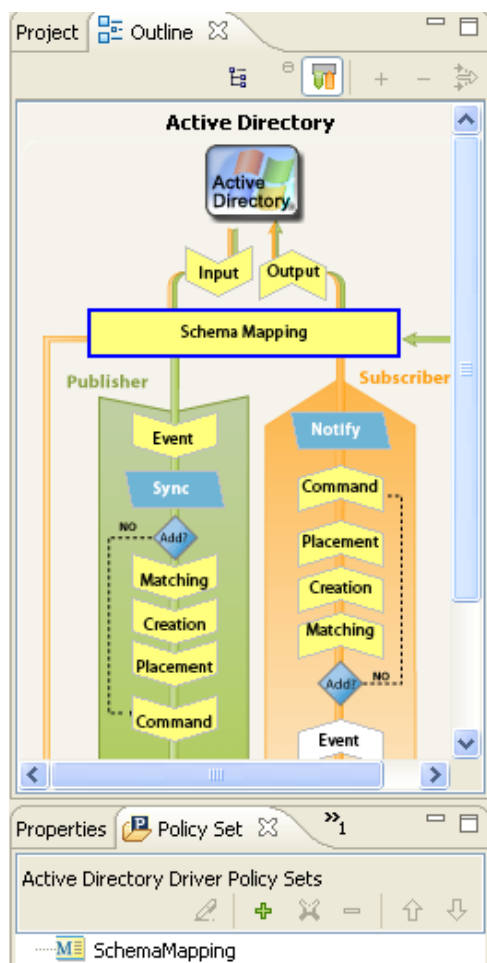
按一下滑鼠右鍵，並選取「編輯」。



規則流程檢視窗

- 1 在開啓的專案中，按一下「大綱」索引標籤。
 - 2 按一下「顯示規則流程」圖示。
 - 3 連按兩下「綱要映射」規則，以啓動「綱要映射」編輯器。
- 或

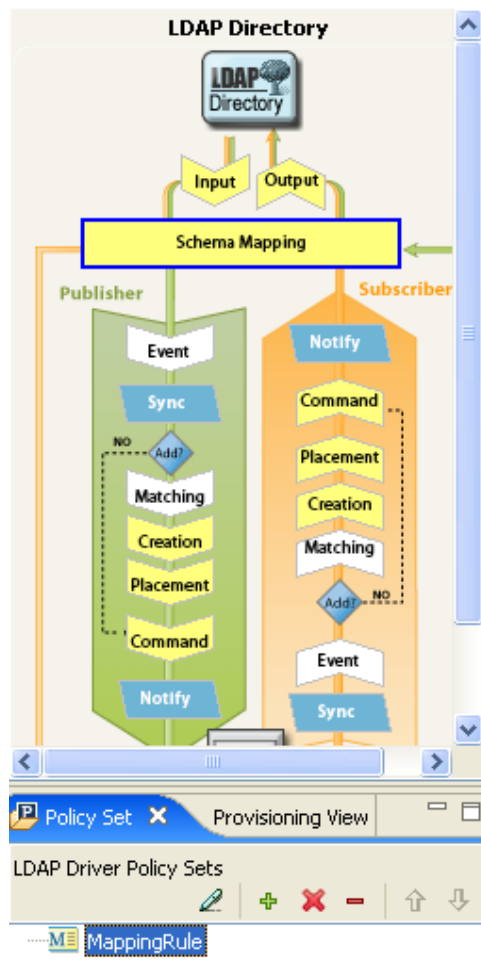
按一下滑鼠右鍵，並選取「編輯規則」，以啓動「綱要映射」編輯器。



規則集檢視窗

- 1 在「規則集」檢視窗中，連按兩下「綱要映射」規則。
或

以滑鼠右鍵按一下「綱要映射」規則，並選取「編輯」。



鍵盤支援

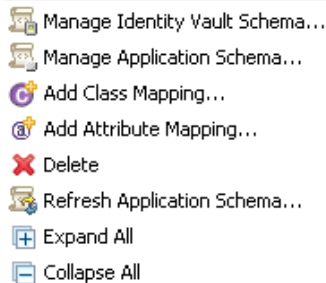
表格 7-1 綱要映射編輯器的鍵盤支援

動作	描述
向上鍵	在「綱要映射」編輯器中，將游標上移。
向下鍵	在「綱要映射」編輯器中，將游標下移。
向左鍵	折疊顯示的資訊
向右鍵	展開顯示的資訊。
Insert	新增類別。
Ctrl+Insert	新增屬性。
Delete	刪除選定項目。
Enter	存取編輯模式。第二次按下 Enter 時可認可變更。
Esc	離開編輯模式。

7.1.2 編輯綱要映射規則

「綱要映射」編輯器可讓您建立並編輯綱要映射規則。若要顯示快顯功能表，請以滑鼠右鍵按一下項目。

特性 7-1 綱要映射編輯器的快顯功能表




- ◆ 「移除或新增類別和屬性」，第 393 頁
- ◆ 「重新整理應用程式綱要」，第 394 頁
- ◆ 「編輯項目」，第 395 頁
- ◆ 「排序項目」，第 395 頁
- ◆ 「管理綱要」，第 395 頁

移除或新增類別和屬性

- ◆ 「移除類別或屬性」，第 393 頁
- ◆ 「新增類別」，第 394 頁
- ◆ 「新增屬性」，第 394 頁

移除類別或屬性

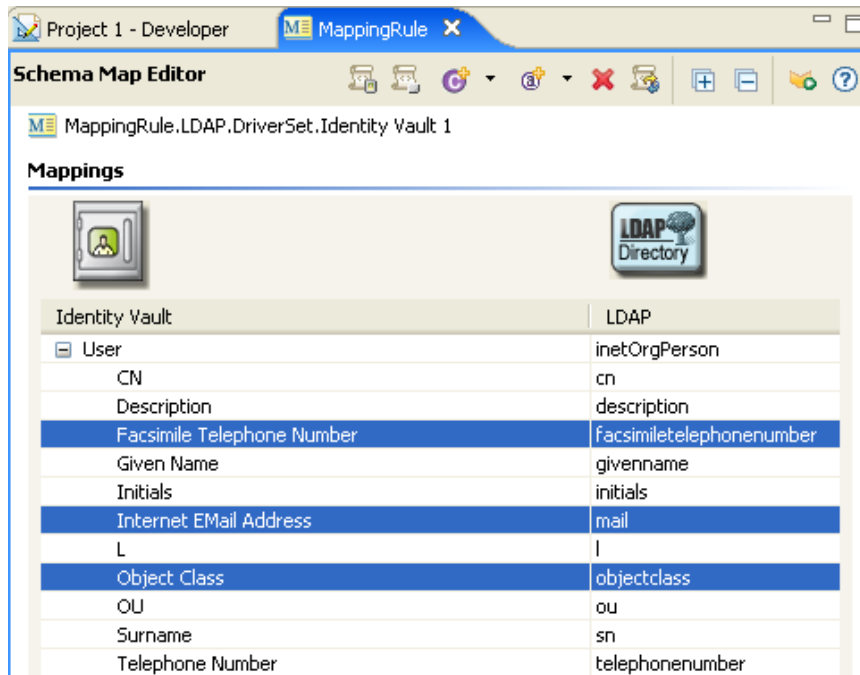
如果不想讓某個類別或屬性映射到已連接系統中的類別或屬性，最佳的作法是將「綱要映射」規則中的類別或屬性完全移除。有三種方法可以新增或移除「綱要映射」規則中的屬性和類別：

- ◆ 選取要移除的類別或屬性，然後按一下滑鼠右鍵，再按一下「刪除」。
- ◆ 選取要移除的類別或屬性，然後按一下右上角的「刪除」圖示 .
- ◆ 選取要移除的類別或屬性，然後按 Delete 鍵。


您可以選取多個類別或屬性，以將其同時刪除。

- 1 按 Ctrl 並以滑鼠選取每個項目。


2 按 Delete 鍵刪除項目。




新增類別

- 1 在「綱要映射」編輯器中按一下滑鼠右鍵，然後按一下「新增類別映射」。
- 或
選取右上角的「新增類別映射」圖示 .
- 2 從 Identity Vault 的下拉式清單中選取要新增的類別。
- 3 從已連接系統的下拉式清單中選取要新增的類別。
- 4 若要儲存變更，請按一下「檔案」>「儲存」。

新增屬性

- 1 在「綱要映射」編輯器中按一下滑鼠右鍵，然後按一下「新增屬性映射」。
- 或
選取右上角的「新增屬性映射」圖示 .
- 2 從 Identity Vault 的下拉式清單中選取要新增的屬性。
- 3 從已連接系統的下拉式清單中選取要新增的屬性。
- 4 若要儲存變更，請按一下「檔案」>「儲存」。

重新整理應用程式綱要

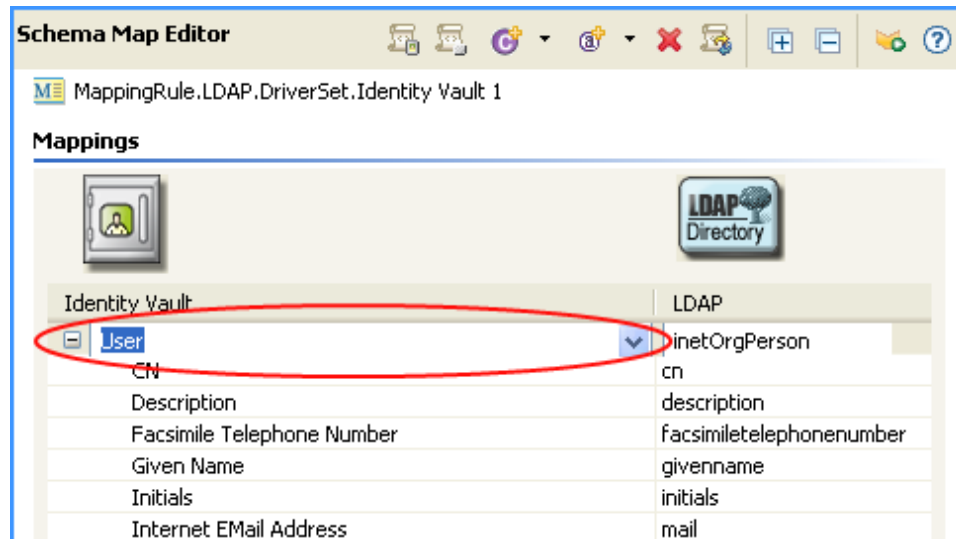
如果你已修改應用程式中的綱要，則這些變更需要反應在「綱要映射」規則中。若要讓新的綱要可供使用，請按一下工具列中的「重新整理應用程式綱要」圖示 .

建立新類別或屬性映射時，您可在已連接應用程式的下拉式清單中看到新的綱要。

編輯項目

若要編輯映射，請連按兩下選定的列。編輯器會就地顯示，可讓您編輯映射。

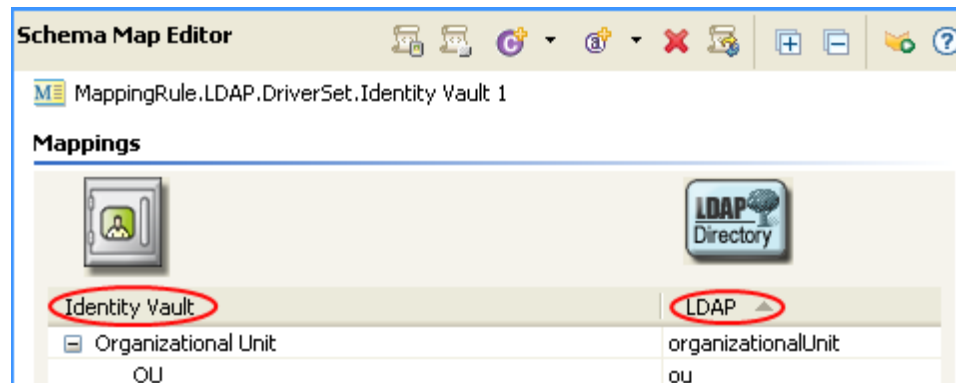
特性 7-2 綱要映射編輯器



排序項目

「綱要」編輯器可讓您根據 Identity Manager 或已連接系統，依照遞增順序排序項目。若要排序，請按一下任一欄的標題。

特性 7-3 綱要映射編輯器的排序項目




管理綱要

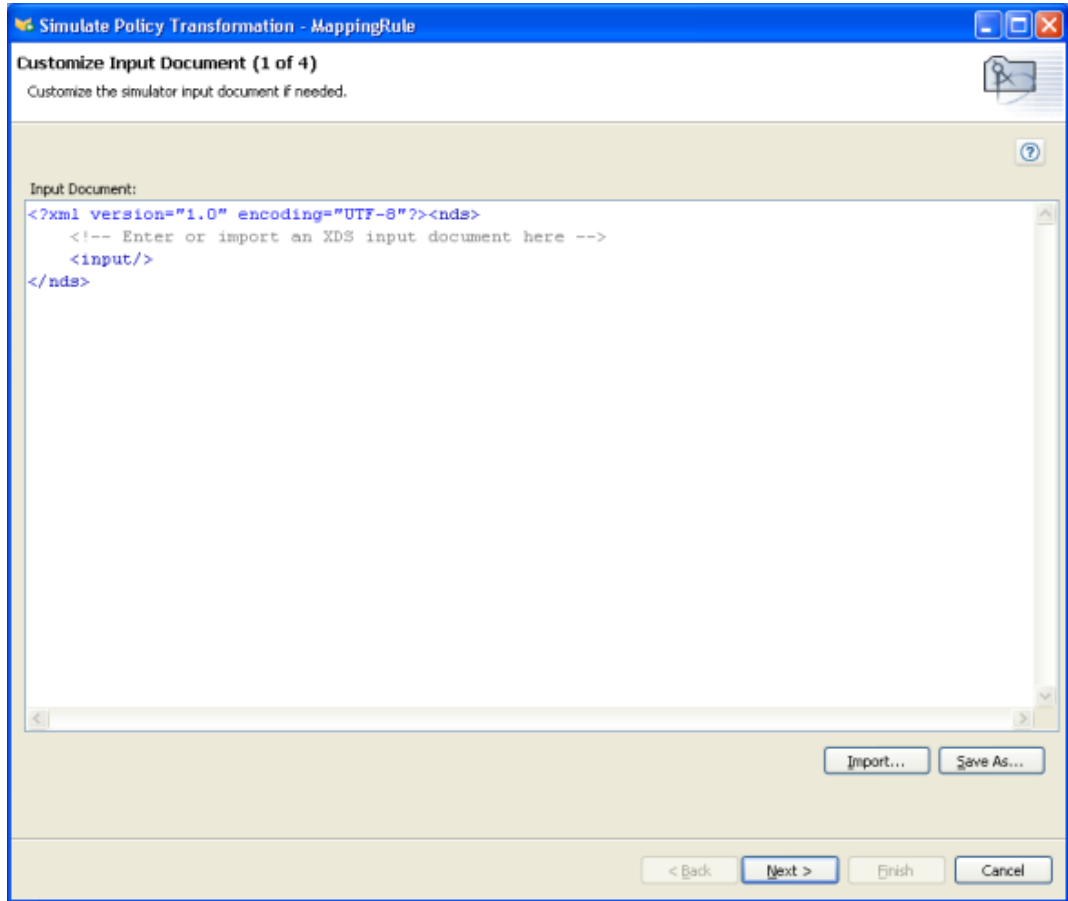
Designer 可讓您管理 Identity Vault 綱要以及任何已連接系統的綱要。您可以輸入綱要並加以修改，還可將變更過的綱要部署回 Identity Vault 或已連接系統中。若要管理 Identity Vault 綱要，請在「綱要映射」編輯器中按一下滑鼠右鍵，然後按一下「管理 Identity Vault 綱要」。若要管理已連接系統的綱要，請在「綱要映射」編輯器中按一下滑鼠右鍵，然後按一下「管理應用程式綱要」。如需如何管理綱要的相關資訊，請參閱《Designer for Identity Manager 3：管理說明》中的「管理綱要」。

7.1.3 測試綱要映射規則

Designer 隨附一套名為「規則模擬器」的工具。該工具可讓您測試規則而不在生產環境中加以實作。您可以透過「綱要映射」編輯器啟動「規則模擬器」，以便在修改規則後加以測試。

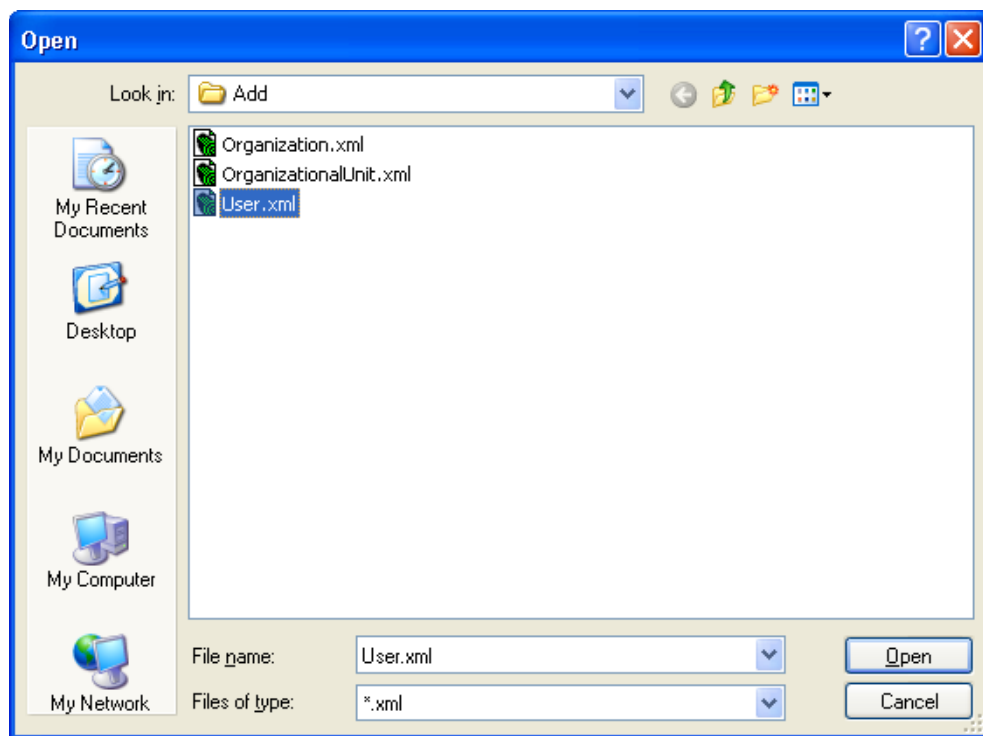
若要存取「規則模擬器」並測試「綱要映射」規則，請執行下列動作：

- 1 按一下工具列中的「啟動規則模擬器」圖示 。
- 2 選取「輸入」以瀏覽至模擬事件的檔案。



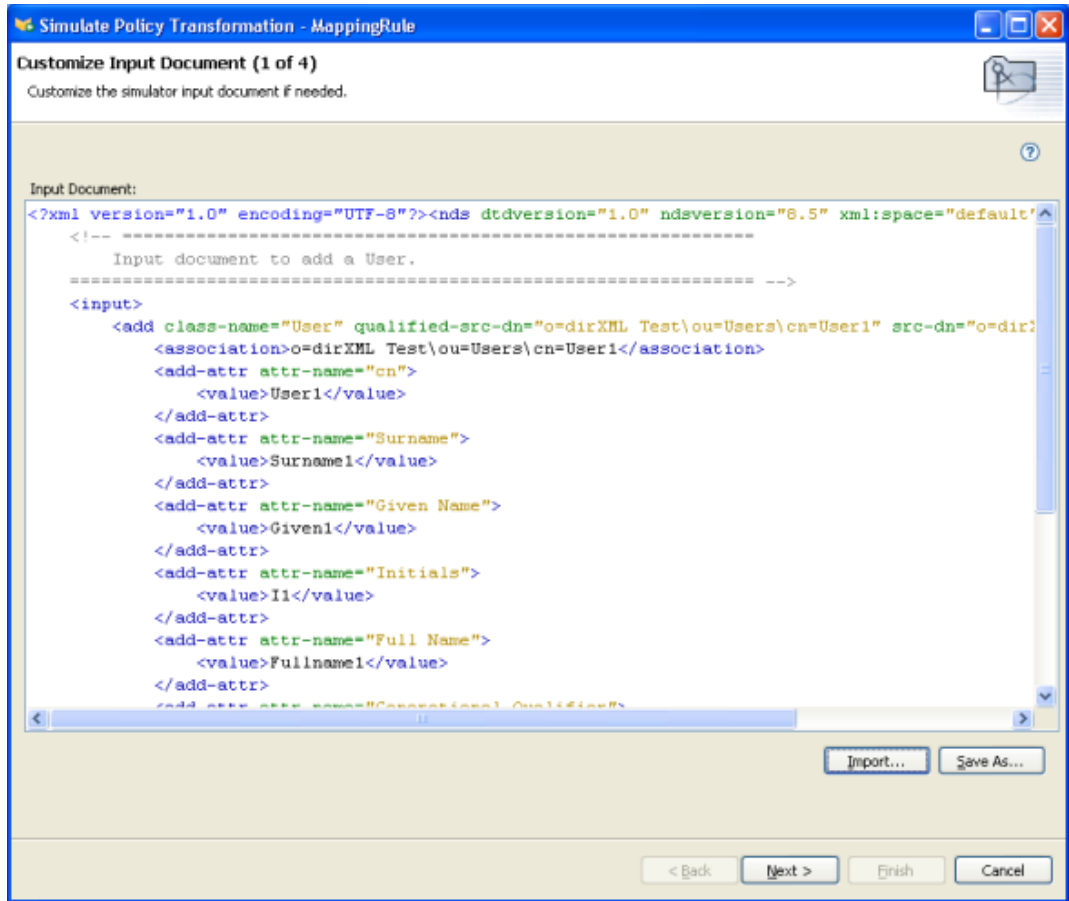
- 3 選取檔案，然後按一下「開啓」。

此範例使用 `com.novell.designer.policy\simulation\add\user.xml` 檔案，該檔案會模擬使用者物件的「新增」事件。

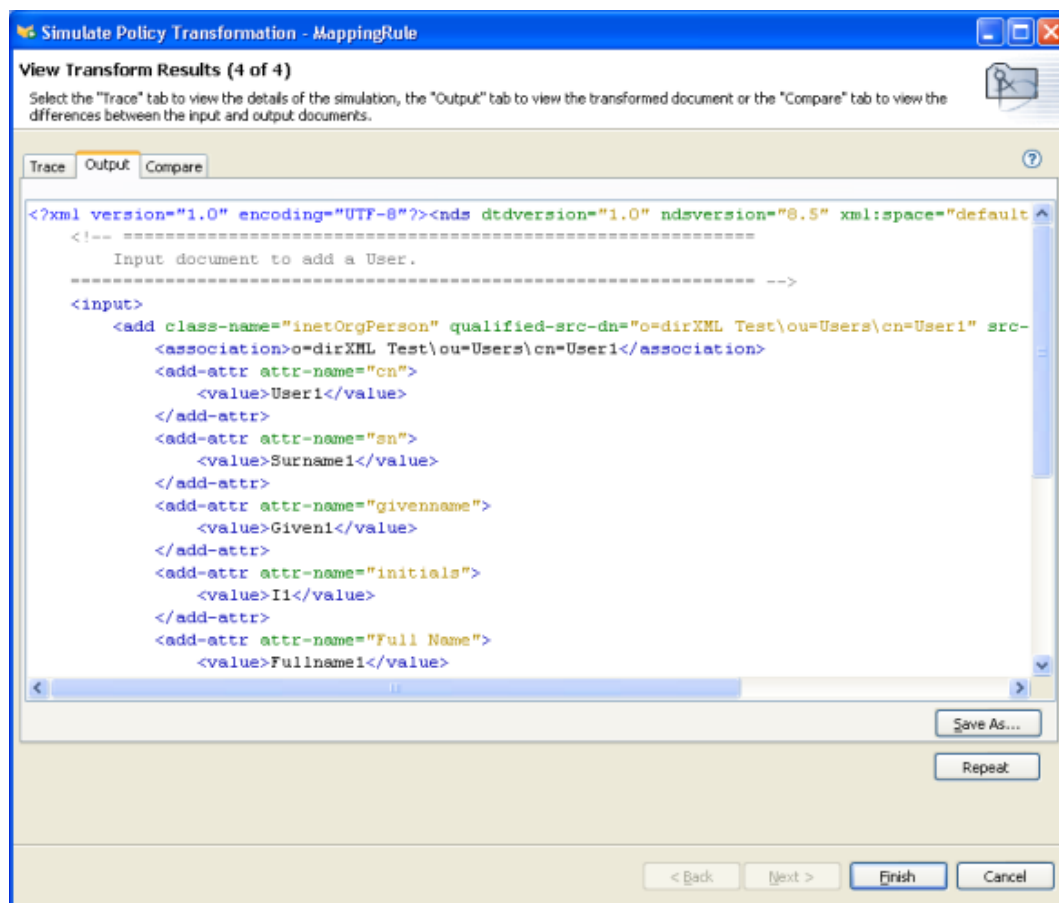


「規則模擬器」會顯示使用者「新增」事件的輸入文件。

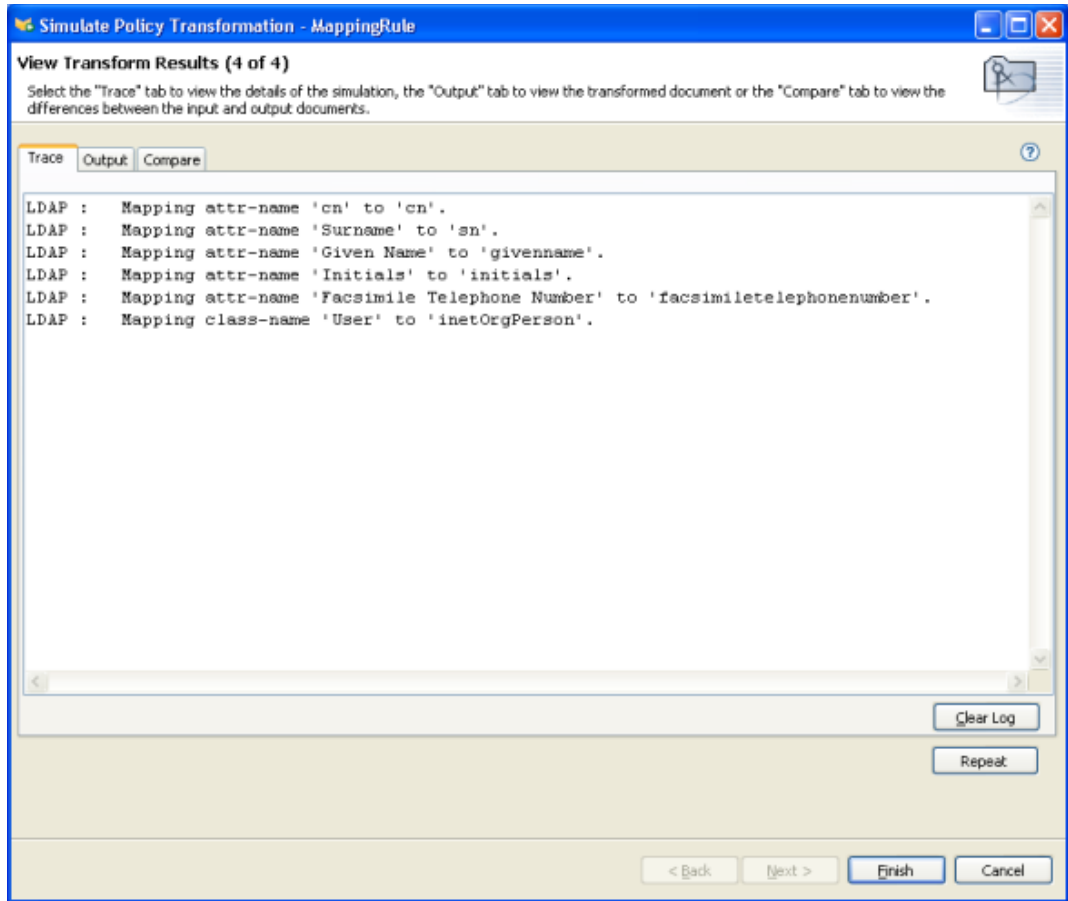
4 按「下一步」開始模擬。



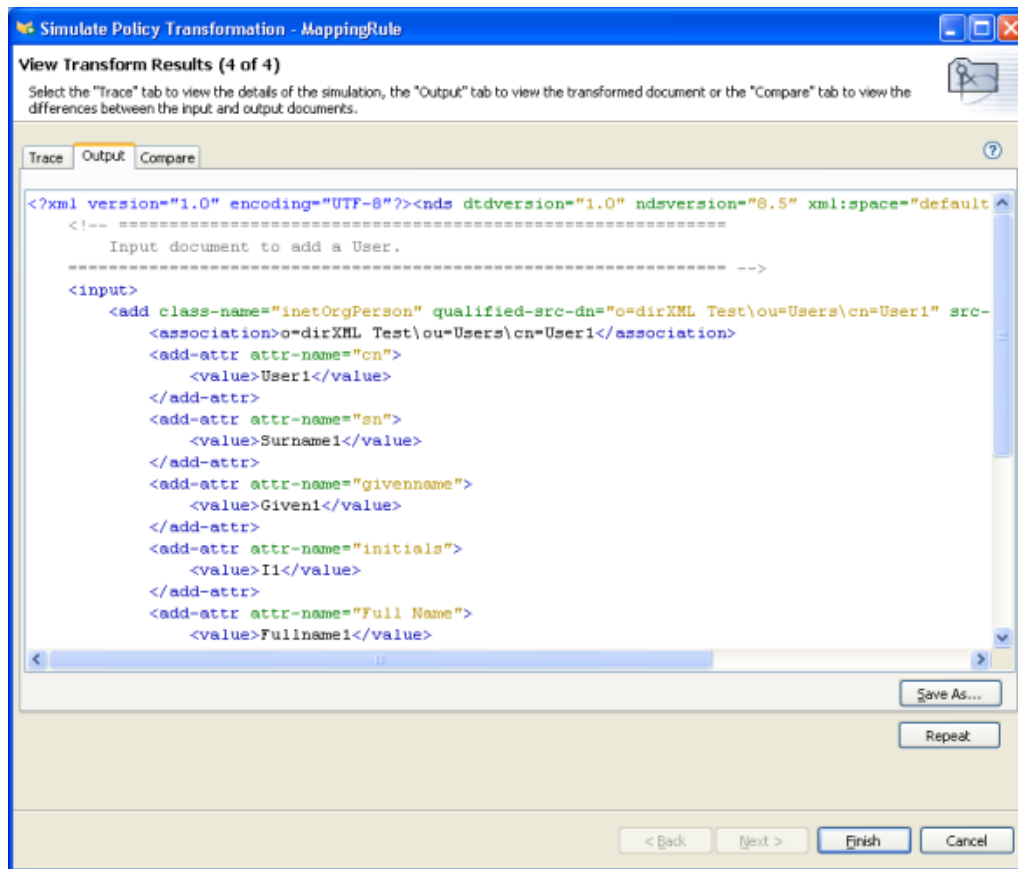
「規則模擬器」會顯示「新增」事件的記錄、輸出文件，以及輸入文件與產生之輸出文件的比較。



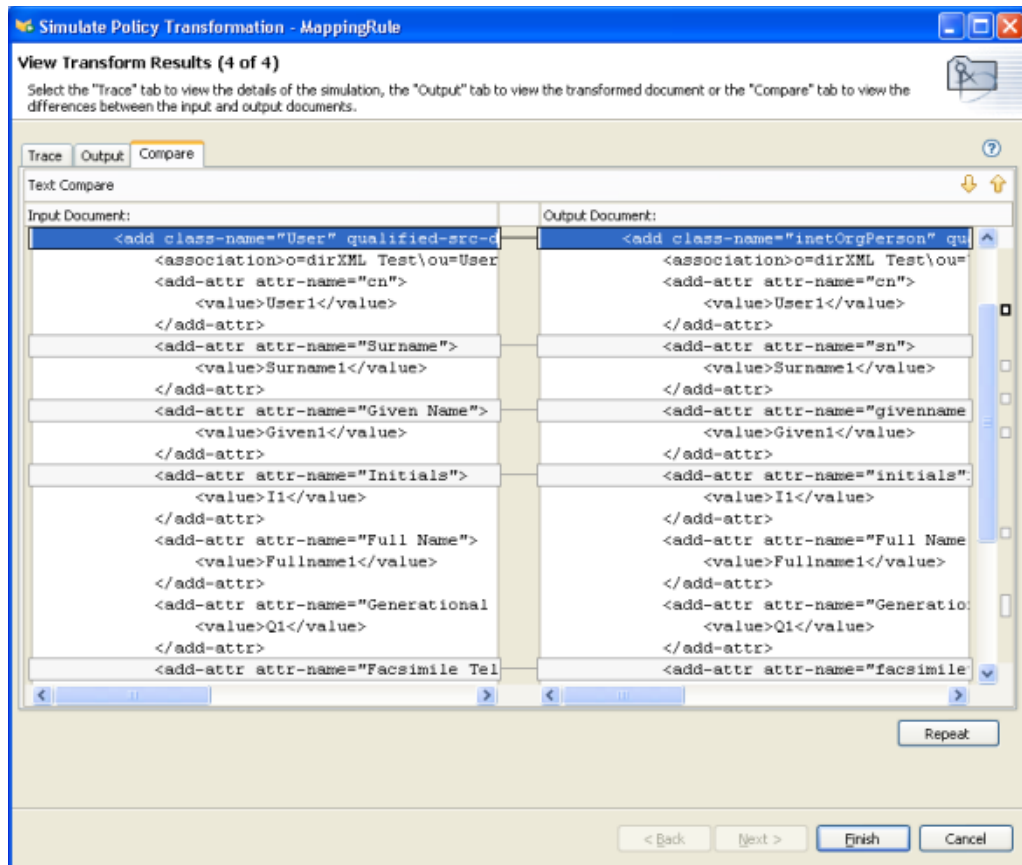
5 選取「追蹤」索引標籤，即可查看您透過 DSTRACE 會看到之「新增」事件的結果。



- 6 選取「輸出」索引標籤，以檢視針對輸入文件執行「綱要映射」規則時產生的輸出文件。在此範例中，輸出文件即為使用者「新增」事件。



7 選取「比較」索引標籤，以比較輸入文件的文字與產生的文件（即輸出文件）。



8 按一下「重複」以選取不同的輸入文件，並查看該事件的結果。

9 當您完成「綱要映射」規則的測試時，請按一下「完成」，關閉「規則模擬器」。

7.1.4 存取綱要映射規則 XML

Designer 可讓您藉由使用 XML 編輯器或文字編輯器來檢視、編輯及驗證 XML。

- ◆ 「[檢視 XML 來源](#)」，第 402 頁
- ◆ 「[編輯 XML 來源](#)」，第 406 頁
- ◆ 「[驗證 XML 來源](#)」，第 408 頁

檢視 XML 來源

您可以 XML 或以 XML 樹狀結構格式檢視「XML 來源」。

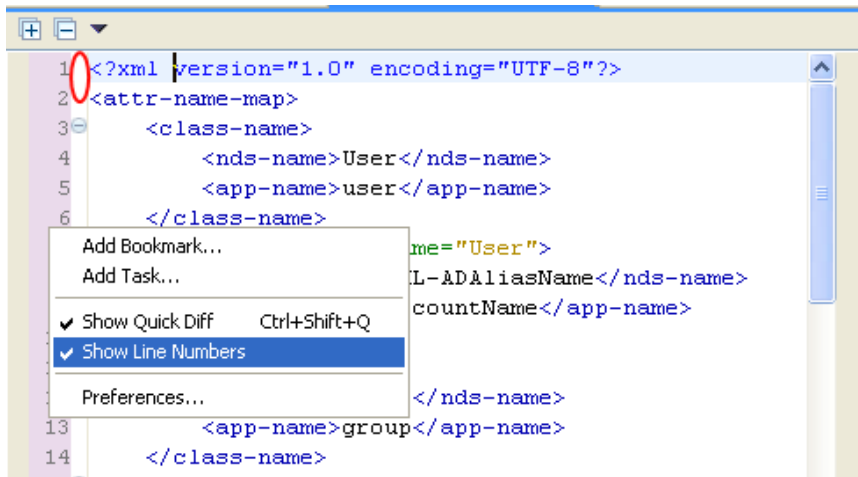
若要開啓「XML 來源」檢視窗，請執行下列動作：

- 1 按一下位於「綱要映射」編輯器之工作空間底部的「XML 來源」。



XML 編輯器會顯示行號。若要查看行號，請以滑鼠右鍵按一下左邊界，然後選取「顯示行號」。

特性 7-4 綱要映射規則的行號



XML 編輯器會依功能展開或折疊 XML。如果有些功能包含大量的 XML，您可以按一下位於左上角的減號圖示，以折疊 XML。若要展開所有的 XML 功能，請按一下位於左上角的加號圖示。

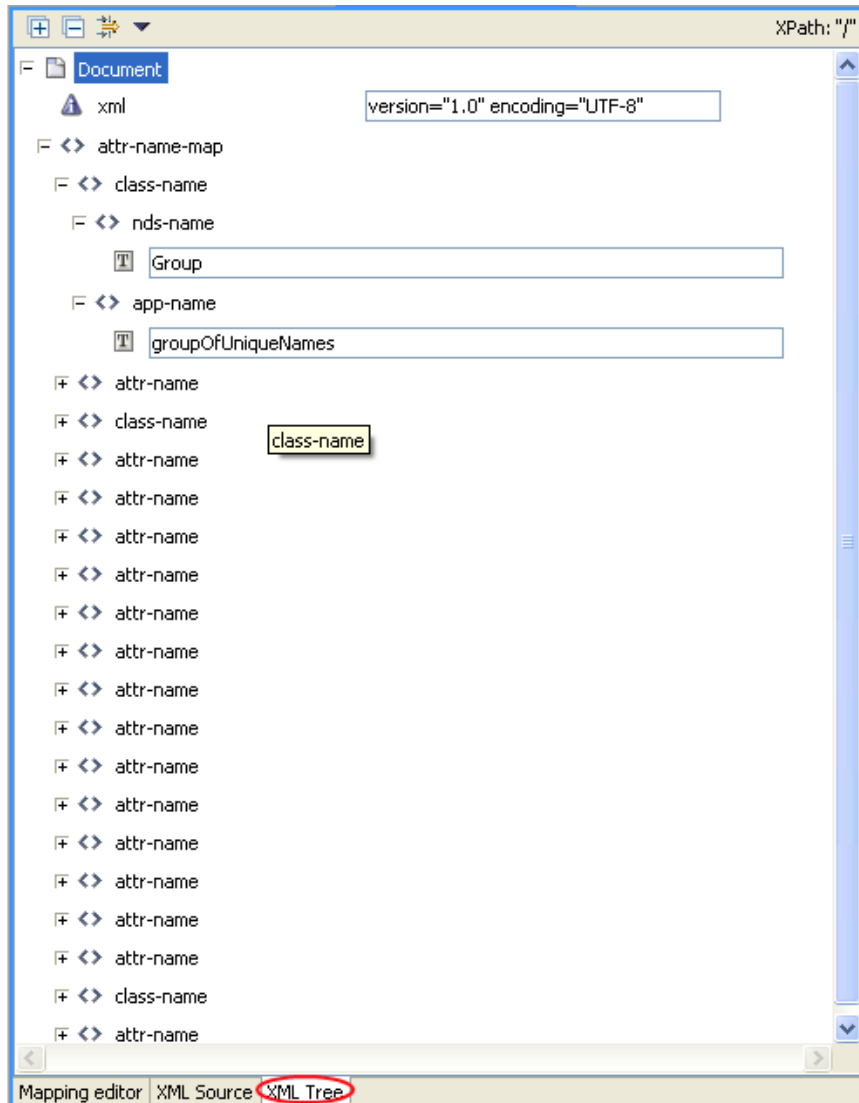
每個元素的左邊界皆擁有其各自的加號或減號圖示。

特性 7-5 綱要映射規則 XML 的加號或減號



若要以樹狀結構格式檢視 XML，請執行下列動作：

- 1 按一下位於「綱要映射」編輯器之工作空間底部的「XML 樹狀結構」。

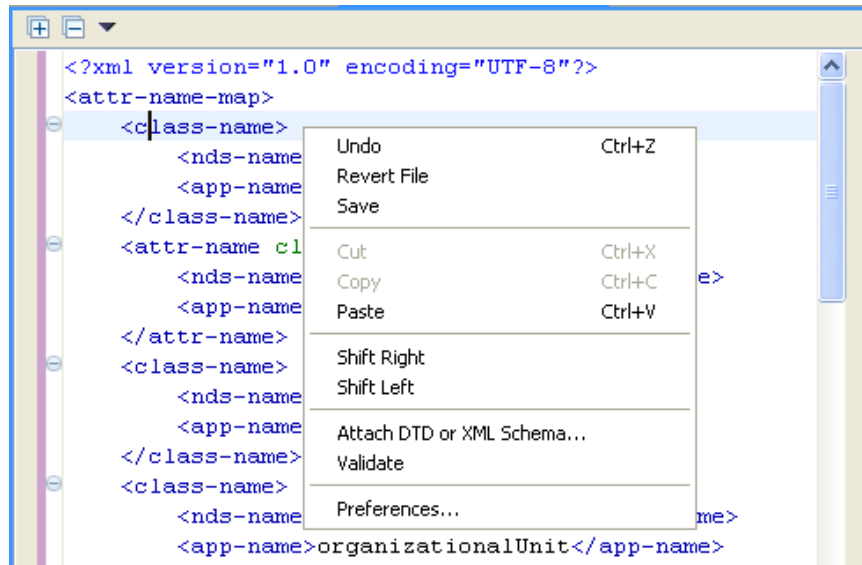


若要查看整個樹狀結構檢視窗，請展開列出的每個項目。

編輯 XML 來源

您可以透過 XML 編輯器編輯 XML。您也可以透過 GUI 介面在此處進行變更。

特性 7-6 編輯綱要映射規則的 XML 來源



載入的預設編輯器與 .xml 檔案類型相關聯。若找不到預設編輯器，則會載入系統文字編輯器。「XML 來源」檢視窗功能是以進行載入的編輯器為基礎。

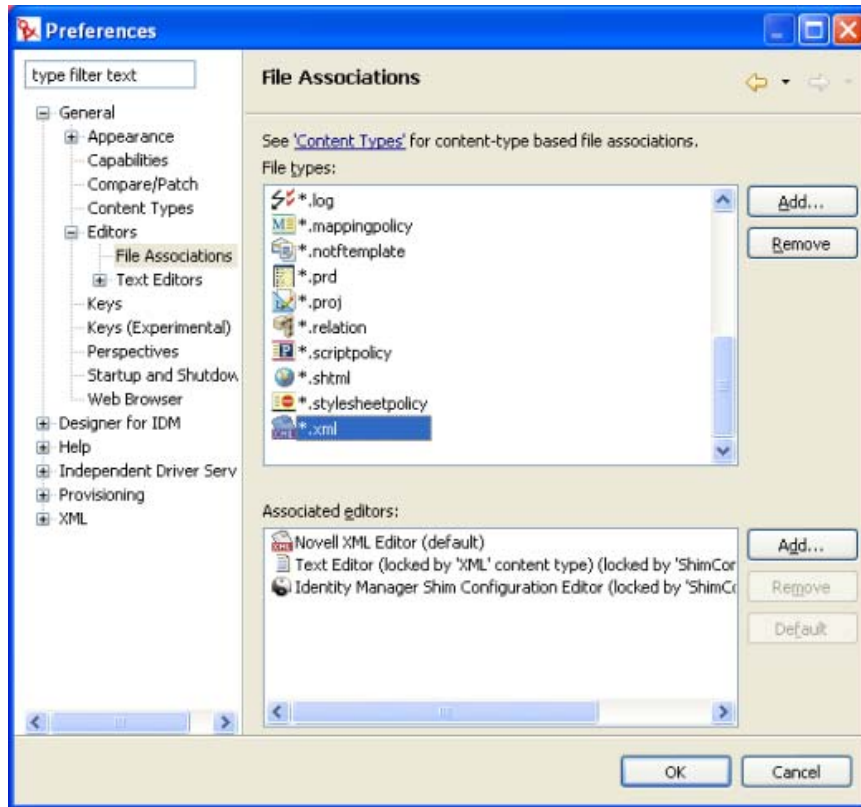
若要顯示 XML 編輯器包含的功能清單，請按一下滑鼠右鍵。

- ◆ 復原：復原上一個動作。
- ◆ 回復檔案 將檔案回復到上一次儲存的版本。
- ◆ 儲存：儲存檔案。
- ◆ 剪下：剪下選定資訊。
- ◆ 貼上：將資訊貼至文件。
- ◆ 向右移：向右縮排。
- ◆ 向左移：向左縮排。
- ◆ 附加 DTD 或 XML 綱要：附加用於驗證規則的 DTD 或 XML 綱要檔案。
- ◆ 驗證：驗證 XML 程式碼。
- ◆ 優先設定：設定 XML 編輯器的優先設定。

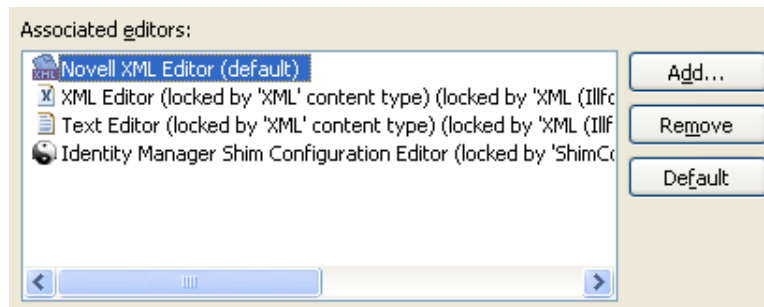
若要為您的來源檢視窗選擇不同的 XML 編輯器，請執行下列動作：

- 1 按一下「主功能表」中的「視窗 > 優先設定」。
- 2 按一下「一般 > 編輯器 > 檔案關聯」。

- 3 選取檔案類型清單中的「*.xml」。



- 4 從「關聯的編輯器」選取所需的編輯器 (如「Novell XML Editor」)。若您要使用的編輯器未出現在清單中，可以按一下「新增」，然後將其新增至清單中。



- 5 按一下「確定」。
- 6 關閉並重新開啓「綱要映射」編輯器。預設編輯器應在「XML 來源」檢視窗中載入。

驗證 XML 來源

XML 編輯器會驗證 XML 程式碼。按一下滑鼠右鍵，然後選取「驗證」。若出現錯誤，則在發生錯誤的該行上會顯示一個紅色的 x。視窗底端的說明提供了該問題的相關資訊。

特性 7-7 驗證綱要映射規則



在此例中，<attr-name> 的結束標籤沒有相符的開始標籤。

7.1.5 綱要映射規則的其他選項

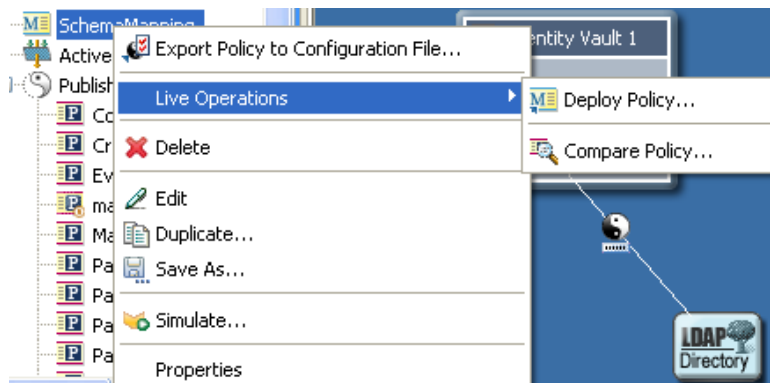
以滑鼠右鍵按一下「綱要映射」規則時，「大綱」檢視窗、「規則流程」檢視窗和「規則集」檢視窗中會出現多個選項。

- 「大綱檢視窗的其他選項」，第 409 頁
- 「規則流程檢視窗的其他選項」，第 409 頁

- ◆ 「規則集檢視窗的其他選項：」，第 411 頁

大綱檢視窗的其他選項

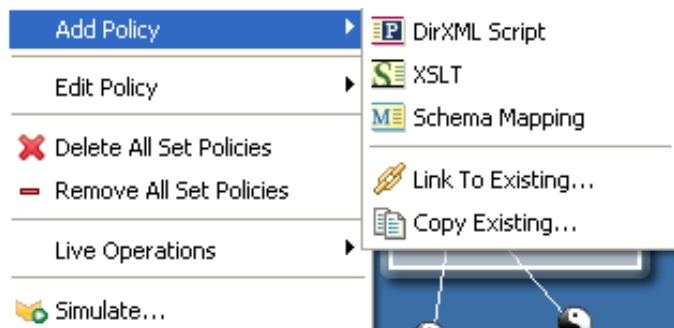
- 1 在「大綱」檢視窗中，以滑鼠右鍵按一下「綱要映射」規則。



- ◆ 將規則輸出至組態檔案：將「綱要映射」規則儲存為 .xml 檔案。
- ◆ 即時操作 > 部署規則：將「綱要映射」規則部署至 Identity Vault。
- ◆ 即時操作 > 比較規則：比較 Designer 中的「綱要映射」規則與 Identity Vault 中的「綱要映射」規則。
- ◆ 刪除：刪除「綱要映射」規則。
- ◆ 編輯：啟動「綱要映射」編輯器。如需相關資訊，請參閱「[編輯綱要映射規則](#)」，第 393 頁。
- ◆ 複製：建立「綱要映射」規則的副本。
- ◆ 另存新檔：將「綱要映射」規則儲存為 .xml 檔案。
- ◆ 模擬：測試「綱要映射」規則。如需相關資訊，請參閱「[測試綱要映射規則](#)」，第 396 頁。
- ◆ 內容：可讓您重新命名「綱要映射」規則。

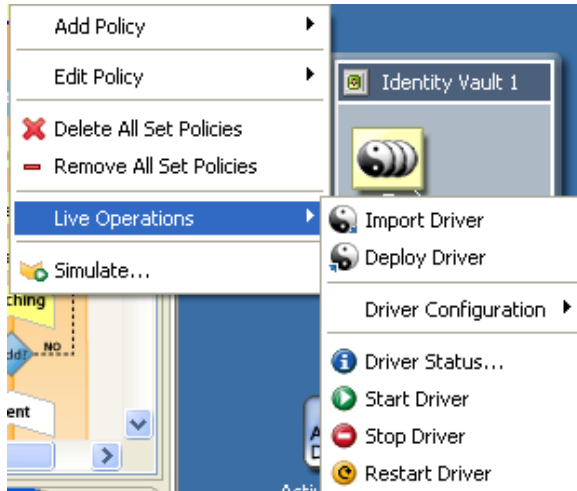
規則流程檢視窗的其他選項

- 1 在「規則流程」檢視窗中，以滑鼠右鍵按一下「綱要映射」規則。



- ◆ 新增規則 > DirXML 程序檔：使用 DirXML® 程序檔新增「綱要映射」規則。
- ◆ 新增規則 > XSLT：使用 XSLT 新增「綱要映射」規則。

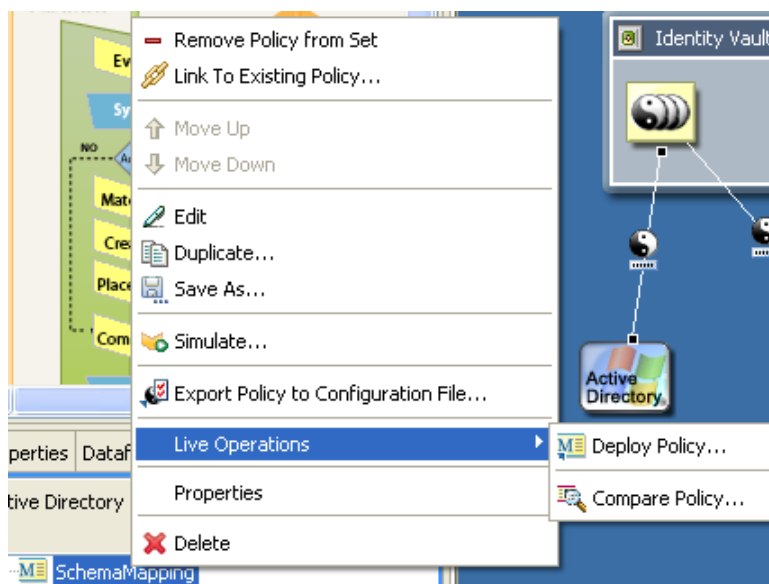
- ◆ 新增規則 > 綱要映射：新增「綱要映射」規則，其中不包含資訊。
- ◆ 新增規則 > 連結至現有規則：可讓您瀏覽並選取現有的「綱要映射」規則，以連結至目前的「綱要映射」規則。
- ◆ 新增規則 > 複製現有規則：可讓您瀏覽至現有的「綱要映射」規則並加以選取，以複製到目前的「綱要映射」規則。
- ◆ 編輯規則 > 綱要映射：啟動「綱要映射」編輯器。如需相關資訊，請參閱「[編輯綱要映射規則](#)」，第 412 頁。
- ◆ 刪除規則集的所有規則：刪除選定規則集中的所有規則。
- ◆ 移除規則集的所有規則：移除選定規則集中的所有規則，但不刪除現有的規則。



- ◆ 即時操作 > 輸入驅動程式：從 Identity Vault 輸入現有的驅動程式。
- ◆ 即時操作 > 部署驅動程式：將現有的驅動程式部署到 Identity Vault 中。
- ◆ 即時操作 > 驅動程式組態 > 輸入屬性：可讓您從 Identity Vault 輸入屬性，並且比較 Identity Vault 和 Designer 中的屬性。
- ◆ 即時操作 > 驅動程式組態 > 部署屬性：可讓您將屬性從 Designer 佈署到 Identity Vault 中，並且比較 Designer 和 Identity Vault 中的屬性。
- ◆ 即時操作 > 驅動程式狀態：顯示驅動程式的狀態。
- ◆ 即時操作 > 啟動驅動程式：啟動驅動程式。
- ◆ 即時操作 > 停止驅動程式：停止驅動程式。
- ◆ 即時操作 > 重新啟動驅動程式：重新啟動驅動程式。
- ◆ 模擬：測試「綱要映射」規則。如需相關資訊，請參閱「[測試綱要映射規則](#)」，第 396 頁。

規則集檢視窗的其他選項：

- 1 在「規則集」檢視窗中，以滑鼠右鍵按一下「綱要映射」規則。



- 移除規則集中的規則：移除規則集中的「綱要映射」規則，但不刪除「綱要映射」規則。
- 連結至現有規則：可讓您瀏覽至另一個「綱要映射」規則，並將其連結至現有的規則。
- 向上移：依照規則的執行順序，將「綱要映射」規則向上移。
- 向下移：依照規則的執行順序，將「綱要映射」規則向下移。
- 編輯：啟動「綱要映射」編輯器。如需相關資訊，請參閱「[編輯綱要映射規則](#)」，第 412 頁。
- 複製：建立「綱要映射」規則的副本。
- 另存新檔：將「綱要映射」規則儲存為 .xml 檔案。
- 模擬：測試「綱要映射」規則。如需相關資訊，請參閱「[測試綱要映射規則](#)」，第 396 頁。
- 將規則輸出至組態檔案：將「綱要映射」規則儲存為 .xml 檔案。
- 即時操作 > 部署規則：將「綱要映射」規則部署至 Identity Vault。
- 即時操作 > 比較規則：比較 Designer 中的「綱要映射」規則與 Identity Vault 中的「綱要映射」規則。
- 內容：可讓您重新命名「綱要映射」規則。
- 刪除：刪除「綱要映射」規則。

7.2 iManager 中的綱要映射規則任務

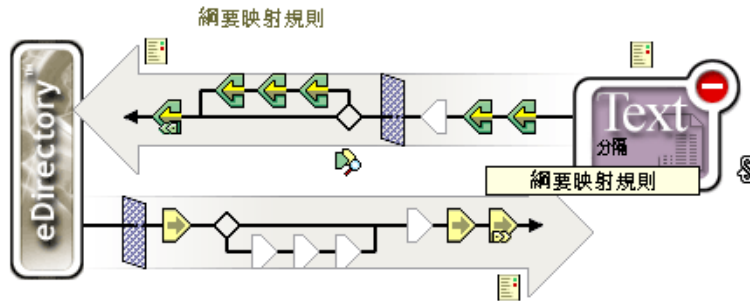
本節包含在 iManager 中執行與「綱要映射」規則相關之常見任務的指示：

- 「[存取綱要映射規則](#)」，第 412 頁

- ◆ 「編輯綱要映射規則」，第 412 頁

7.2.1 存取綱要映射規則

- 1 在 iManager 中展開「Identity 管理」角色，然後按一下「Identity Manager 概觀」。
- 2 選取「搜尋整個網路樹」或「在容器中搜尋」以搜尋某個「驅動程式」集，然後按一下「搜尋」。
- 3 按一下要管理「綱要映射規則」的驅動程式。此時會開啓「Identity Manager 驅動程式概觀」頁面。



- 4 按一下「綱要映射規則」。
- 5 按一下「編輯」。

7.2.2 編輯綱要映射規則

編輯「綱要映射」規則分為兩個不同的部份。第一，編輯規則集中的規則佈置。第二，透過「綱要映射」編輯器來編輯規則本身。

規則佈置

按一下「綱要映射規則」時會出現內含選項的視窗。

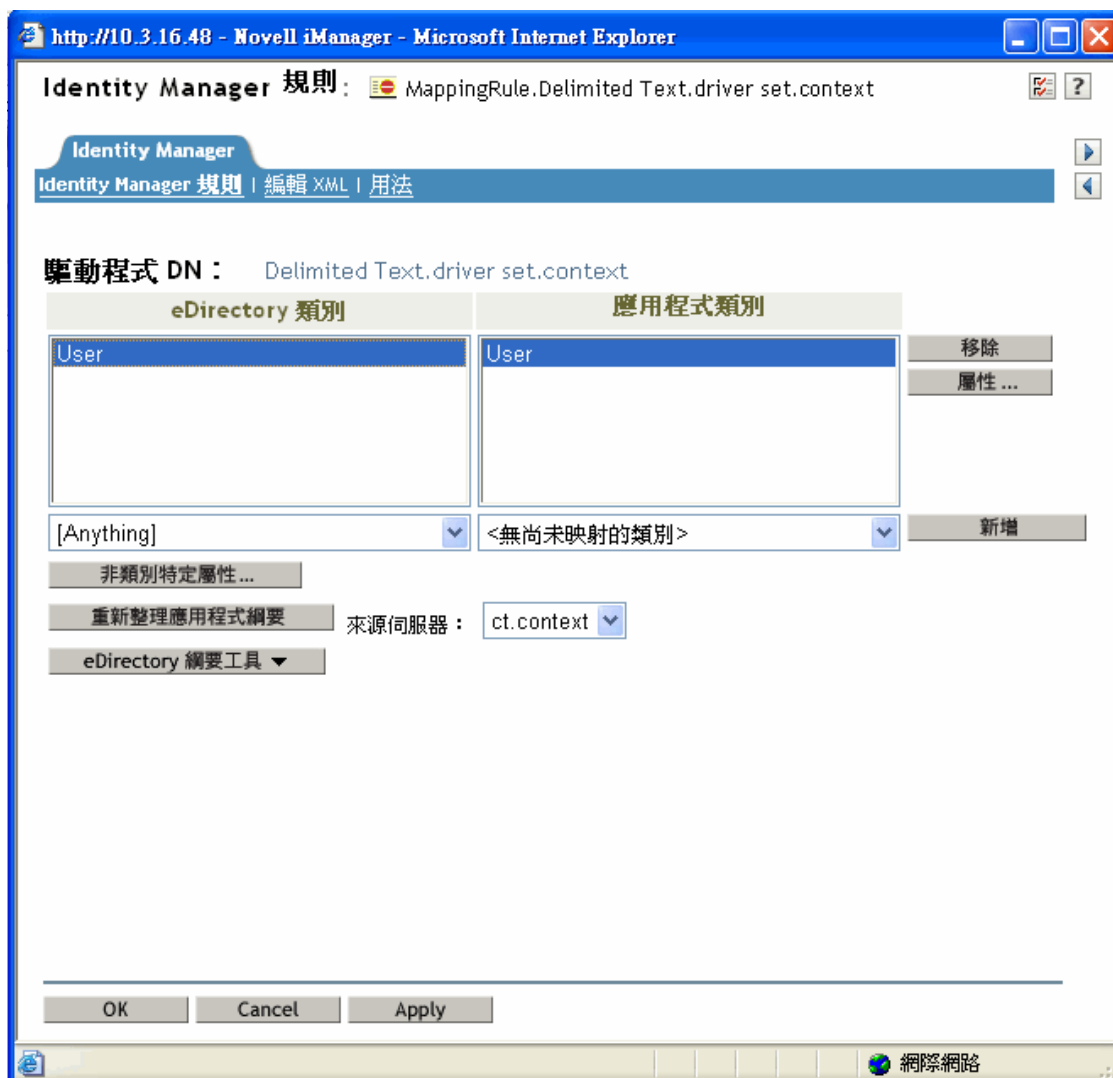


這些選項可讓您將正在使用的規則定位。下表會說明各選項。

選項	描述
將規則向上移	如果規則超過一個，則會將選定的規則向上移。
將規則向下移	如果規則超過一個，則會將選定的規則向下移。
插入	將新的規則或現有規則插入列出的規則中。
移除	移除選定的規則，但不將其從規則集中刪除。
編輯	啟動「綱要映射」編輯器。
重新命名	重新命名選定的規則。
刪除	刪除選定的規則。

綱要映射編輯器

「綱要映射」編輯器是一個完整的圖形化介面，可供建立並管理綱要映射規則。「綱要映射」編輯器會使用 XML 建立規則。



「綱要映射」編輯器有三個索引標籤：

- ◆ 「Identity Manager 規則」，第 414 頁
- ◆ 「編輯 XML」，第 414 頁
- ◆ 「用法」，第 414 頁

Identity Manager 規則

包含最多的資訊，同時也是透過 GUI 介面編輯規則的地方。您可以在「綱要映射」編輯器中進行下列任務：

移除類別和屬性	選取要移除的類別或屬性，然後按一下「移除」。
新增類別	從下拉式清單選取 eDirectory 類別，然後從下拉式清單中選取「應用程式」類別。選取項目後，請按一下「新增」，然後按一下「套用」以儲存變更。
新增屬性	選取要新增的屬性類別，然後按一下「屬性」。從下拉式清單選取 eDirectory 屬性，然後從下拉式清單中選取「應用程式」屬性。選取項目後，請按一下「新增」，然後按一下「確定」以儲存變更。
列出非特定的類別屬性	如果有些屬性未與類別相關聯，請按一下「非特定的類別屬性」圖示，以及所有列出的那些屬性。
重新整理應用程式綱要	如果應用程式的綱要已經變更，請按一下「重新整理應用程式綱要」圖示。精靈會聯絡「已連接系統」伺服器，以取回新的綱要。綱要更新後會列於下拉式清單中。
使用 eDirectory 綱要工具	<ul style="list-style-type: none">◆ 新增屬性 – 將現有的屬性新增至選定的類別。◆ 建立屬性 – 建立新的屬性。◆ 建立類別 – 建立新的類別。◆ 刪除屬性 – 刪除選定的屬性。◆ 刪除類別 – 刪除選定的類別。◆ 重新整理 eDirectory 綱要 – 對 eDirectory 綱要进行變更後，請按一下「重新整理 eDirectory 綱要」，而下拉式清單中的資訊便會更新。

警告：請勿刪除 Identity Vault 中正在使用的任何類別或屬性，這可能會造成物件變成未知物件。

編輯 XML

按一下「啟用 XML 編輯」可讓您編輯「DirXML 程序檔」規則。對「DirXML 程序檔」進行所需的變更，然後按一下「套用」以儲存變更。

用法

向您顯示目前正在參考此規則之驅動程式的清單。該清單只參考此規則之驅動程式集中的規則。如果此規則是從不同的驅動程式集參考，那些參考便不會出現在此處。